KLASIFIKASI *HOAX* PADA BERITA KESEHATAN BERBAHASA INDONESIA DENGAN MENGGUNAKAN METODE *MODIFIED K-NEAREST NEIGHBOR*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Andre Rino Prasetyo
NIM: 145150207111125

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

KLASIFIKASI *HOAX* PADA BERITA KESEHATAN BERBAHASA INDONESIA DENGAN MENGGUNAKAN METODE *MODIFIED K-NEAREST NEIGHBOR*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Andre Rino Prasetyo
NIM: 145150207111125

Skripsi ini telah diuji dan dinyatakan lulus pada
31 Juli 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Indriati, S.T, M.Kom	Putra Pandu Adikara, S.Kom, M.Kom
NIP. 19831013 201504 2 002	NIP. 19850725 200812 1 002

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 31 Juli 2018

Andre Rino Prasetyo

NIM: 145150207111125

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul "Klasifikasi *Hoax* Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode *Modified K-Nearest Neighbor*" ini dapat terselesaikan. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Indriati, S.T, M.Kom dan Bapak Putra Pandu Adikara, S.Kom, M.Kom selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak dr. Wisnu Wijanarko, Sp.An-KIC dan Bapak dr. Samsul Arif yang bertindak sebagai pakar dalam mengkategorikan jenis berita dalam skripsi ini, atas segala ilmu dan waktu yang diberikan kepada penulis.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., Bapak Ir. Heru Nurwarsito, M.Kom, Bapak Suprapto, S.T, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan I, Wakil Dekan II dan Wakil Dekan III Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, Bapak Agus Wahyu Widodo, S.T, M.Cs dan Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc selaku Ketua Jurusan, Ketua Program Studi dan Sekretaris Program Studi Teknik Informatika.
5. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
6. Seluruh dosen, staff serta teman-teman Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 31 Juli 2018

Penulis

andrerinoprasetyo@gmail.com

ABSTRAK

Andre Rino Prasetyo, Klasifikasi *Hoax* Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode *Modified K-Nearest Neighbor*

Dosen Pembimbing: Ibu Indriati, S.T, M.Kom dan Bapak Putra Pandu Adikara, S.Kom, M.Kom

Berita merupakan sumber informasi mengenai kejadian terkini yang mana dapat diketemukan pada surat kabar, televisi, internet dan media lainnya. Saat ini berita-berita yang disebarkan seringkali tanpa menuliskan sumbernya secara jelas terutama jenis berita mengenai kesehatan, hal tersebut yang dapat mengakibatkan salah penafsiran karena berita tersebut belum tentu benar atau salah sehingga dibutuhkan suatu sistem cerdas untuk mengklasifikasikan berita kesehatan tersebut apakah termasuk dalam kategori *hoax* atau fakta. Proses klasifikasi *hoax* akan menggunakan beberapa tahap mulai dari preprocessing yang terdiri dari tokenisasi dan *filtering*. Dilanjutkan dengan proses pembobotan kata dan cosine similarity hingga proses klasifikasi dengan menggunakan metode *Modified K-Nearest Neighbor*. Hasil yang diperoleh berdasarkan implementasi dan pengujian menghasilkan nilai *k* terbaik berjumlah 4, *precision* sebesar 0,83 *recall* sebesar 0,75 *f-measure* sebesar 0,79 dan akurasi sebesar 75%. Hasil pengujian tersebut didapat karena konten berita kesehatan yang digunakan masih terlalu umum, banyak juga kata-kata yang tidak baku dan penentuan nilai *k-value*s yang digunakan sangat berpengaruh terhadap baik tidaknya proses klasifikasi dokumen berita kesehatan.

Kata kunci: berita kesehatan, klasifikasi, *hoax*, *Modified K-Nearest Neighbor*

ABSTRACT

Andre Rino Prasetyo, Klasifikasi *Hoax* Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode *Modified K-Nearest Neighbor*

Supervisors: Ibu Indriati, S.T, M.Kom and Bapak Putra Pandu Adikara, S.Kom, M.Kom

News is a source of information about current events which can be found in newspapers, television, the internet and other media. Currently the news that is disseminated often without writing the source clearly, especially the type of news about health, it can lead to misinterpretation because the news is not necessarily true or wrong so it takes a smart system to classify health news is whether included in the category of hoax or fact. The hoax classification process will use several stages ranging from preprocessing consisting of tokenisasi and filtering. Continued with word-weighting process and cosine similarity to classification process using Modified K-Nearest Neighbor method. The results obtained based on the implementation and testing resulted in the best value of k amounted to 4, precision of 0,83 recall of 0,75 f-measure of 0,79 and the accuracy of 75%. The test results obtained because the health news content used is still too common, many non-standard words and the determination of k-values used are very influential on whether or not the process of classification of health news documents.

Keywords: health news, classification, hoax, Modified K-Nearest Neighbor

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dewasa ini perkembangan teknologi semakin pesat. Teknologi saat ini yang sedang berkembang pesat adalah internet. Berbicara tentang internet tentu tak lepas dari istilah jejaring sosial yang mana banyak dari media massa menggunakannya sebagai wadah persebaran berita. Namun dalam praktiknya suatu berita di jejaring sosial tidak sepenuhnya sesuai fakta. Sekarang banyak beredar berita yang hanya menyampaikan info dari satu sisi, yang mana sisi negatif lah yang selalu diincar para jurnalis agar menarik minat para pembaca. Selain hal tersebut ada lagi masalah yang sedang dilanda dunia saat ini yakni munculnya berita *hoax*, di dalamnya adalah berita yang memuat informasi palsu dan tidak dapat dipertanggungjawabkan serta bertujuan untuk meyakinkan *netizen* untuk terkecoh dan percaya terhadap berita *hoax* tersebut. *Hoax* mampu memengaruhi banyak orang dengan menodai suatu citra dan kredibilitas (Chen et al, 2014).

Pada tahun 2016 lalu terdapat 300-an akun yang sudah diblokir terkait konten penyebaran informasi *hoax*, isu SARA dan provokasi (Pasaribu, 2016). Penyebaran *hoax* tersebut ada motifnya, entah itu motif politik ataupun motif ekonomi. Kalau motif ekonominya yakni bagaimana semakin sering dikunjungi halaman, masuk halaman mereka maka akan menambah keuntungan secara ekonomis bagi mereka. Dengan jumlah persebaran berita yang saat ini sangat beragam dan banyak sekali, maka akan sulit dalam menentukan suatu berita termasuk *hoax* atau benar adanya. Terutama berita mengenai kesehatan yang banyak sekali disebar melalui grup di beberapa jejaring sosial bahkan situs yang *valid* pun tak luput dari persebaran berita kesehatan yang tidak dapat dipertanggungjawabkan keasliannya. Perlu dilakukan klarifikasi dengan sumber lainnya untuk menentukan apakah suatu berita termasuk ke dalam kriteria *hoax* atau fakta.

Dalam menentukan isi berita kesehatan yang dimuat, apakah suatu berita dapat dikategorikan sebagai *hoax* atau fakta cukup sulit jika hanya melihat dari satu sumber saja. Jika dilihat dengan kasat mata, pesan yang singkat dan terkesan kurang rapi pembaca bisa saja mengklasifikasikan berita tersebut *hoax.* Bahkan jika konten pada berita tersebut terdapat pendapat ahli kesehatan yang disertakan namanya, pembaca dapat menyimpulkan bahwa berita tersebut adalah fakta. Namun dalam kenyataannya kedua studi kasus di atas tidak bisa benar-benar menjadi acuan dalam melabeli jenis suatu berita. Terdapat beberapa ciri khas dari berita kesehatan yang palsu, pertama adalah ketidakjelasan asal usul informasinya, kedua yakni dengan mencantumkan instansi atau nama seseorang yang dikatakan ahli di bidang tersebut yang mengakibatkan beberapa orang langsung percaya bahwa itu adalah berita fakta. Ketiga bisa juga dilihat dari kualitas gambar yang dimuat di dalamnya yang biasanya beresolusi rendah dan bahkan tidak sesuai dengan konteks berita yang disajikan.

Masyarakat seringkali kurang inisiatif dalam membaca suatu berita dari sudut pandang yang luas seperti mencari kebenaran berita yakni membandingkannya

dengan berita di situs lainya ataupun media lain. Alasan yang paling masuk akal adalah ketika membandingkan berita satu dengan berita lain yang dan berkaitan sangat membuang waktu karena derasnya arus informasi menyebabkan para pembaca harus kerja keras mencari referensi sebanyak mungkin. Berdasarkan alasan tersebut, kebutuhan akan sistem cerdas serta akurasi dari klasifikasi berita kesehatan diperlukan suatu perangkat lunak bantu yang akan dikembangkan dalam penelitian ini.

Sebelumnya telah dilakukan penelitian untuk mengklasifikasikan kategori berita *hoax* atau fakta dengan menggunakan tiga algoritme yakni *Naïve Bayes*, *Support Vector Machine* dan *C4.5* untuk dibandingkan hasilnya (Rasywir & Purwarianti, 2015). Hasil penelitian dari 22 topik berita tersebut membuktikan bahwa metode *Naïve Bayes* menghasilkan akurasi dengan nilai terbaik yakni sebesar 91.36% dibandingkan dengan *Support Vector Machine* dan *C4.5*. Disebutkan bahwa penghilangan *stopword* memberikan hasil akurasi lebih tinggi sebesar 1.13% secara rata-rata, serta fitur tanpa *stemming* dapat memberikan akurasi yang lebih baik.

Penggunaan metode *Naïve Bayes* dalam klasifikasi teks telah banyak digunakan karena memang terbukti memperoleh hasil akurasi yang tinggi. Dalam penelitian mengenai klasifikasi teks homograf yang membandingkan metode antara *Naïve Bayes* dengan *K-Nearest Neighbor* juga telah disebutkan bahwa metode *Naïve Bayes* menghasilkan nilai *f-measure* lebih tinggi daripada *K-Nearest Neighbor* yakni dengan rata-rata 89%, karena *K-Nearest Neighbor* sangat bergantung pada pemilihan nilai *k* (Anggono et al, 2009). Lalu pada penelitian yang dilakukan oleh Palinoan dan Wijono (2014) mengenai klasifikasi dokumen berbahasa Jawa yang menggunakan metode *K-Nearest Neighbor* menghasilkan akurasi tinggi dengan nilai *k* sebesar 4. Pengujiannya menggunakan *cross validation* sejumlah 3 *fold*, dari 40 dokumen percobaan dan 4 jenis kategori. Penelitian tersebut memperoleh hasil akurasi yang sangat tinggi yakni sebesar 95%.

Berdasarkan studi kepustakaan di atas, penelitian tentang klasifikasi berita kesehatan berbahasa Indonesia akan menggunakan metode *Modified K-Nearest Neighbor* yang merupakan pengembangan dari metode *K-Nearest Neighbor* yang jarang sekali untuk klasifikasi teks dan rata-rata digunakan untuk klasifikasi penyakit. *Modified K-Nearest Neighbor* disini melakukan perhitungan *cosine distance* dan perhitungan nilai validitas pada semua data latih lalu melakukan perhitungan *weight voting* pada semua data uji menggunakan validitas data (Parvin et al, 2010). Dengan data yang digunakan berupa berita dengan topik kesehatan saja maka metode *Modified K-Nearest Neighbor* ini diharapkan dapat memberikan hasil yang lebih baik dibandingkan dengan penelitian sebelumnya yang dilakukan oleh Rasywir dan Purwarianti (2015).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, dapat diperoleh rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan algoritme *Modified K-Nearest Neighbor* ke dalam sistem klasifikasi *hoax* pada berita kesehatan berbahasa Indonesia?
2. Bagaimana tingkat akurasi dari hasil sistem klasifikasi *hoax* pada berita kesehatan berbahasa Indonesia dengan menggunakan *precision*, *recall, f-measure* dan akurasi?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritme *Modified K-Nearest Neighbor* ke dalam sistem klasifikasi *hoax* pada berita kesehatan berbahasa Indonesia.
2. Mengetahui tingkat akurasi dari hasil sistem klasifikasi *hoax* pada berita kesehatan berbahasa Indonesia dengan menggunakan *precision*, *recall, f-measure* dan akurasi.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah dapat berguna bagi penulis maupun pembaca, selengkapnya sebagai berikut:

1. Bagi penulis

   Memahami penerapan algoritme *Modified K-Nearest Neighbor* dalam klasifikasi berita tentang kesehatan sehingga dapat mendapatkan hasil dan akurasi yang tepat.

2. Bagi pembaca

   Memudahkan pembaca dalam mengetahui suatu berita kesehatan apakah termasuk dalam kategori *hoax* atau fakta yang benar adanya secara lebih cepat sehingga dalam membaca berita tersebut diharapkan untuk diklarifikasi sebelum menerima informasi yang dicantumkan.

1.5 Batasan Masalah

Penelitian ini dilakukan dengan beberapa batasan masalah agar pembahasan nantinya akan jelas kemana arah tujuannya, sebagai berikut:

1. Data sampel yang digunakan merupakan berita yang diperoleh dari beberapa *hoax buster* dan portal berita lalu kedua data tersebut divalidasi kebenarannya oleh pakar sehingga pemberian label *hoax* dan fakta bisa didapatkan.
2. Sistem hanya dapat melakukan klasifikasi untuk mengetahui kategori kebenaran berita.
3. Pemrosesan teks tidak menggunakan *stemming.*
4. Hasil keluaran yang diperoleh nantinya hanya terdapat pernyataan *hoax* dan fakta.

1.6 Sistematika Pembahasan

Urutan penulisan dan deskripsi singkat pada penelitian ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab pertama akan membahas mengenai latar belakang penulisan, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, serta sistematika pembahasan dalam mengklasifikasi berita kesehatan dengan algoritme *Modified K-Nearest Neighbor*.

BAB 2 LANDASAN KEPUSTAKAAN

Bab kedua akan membahas dasar teori atau kajian pustaka seperti pengertian berita, *hoax*, *preprocessing,* pembobotan kata*, cosine similarity,* algoritme *Modified K-Nearest Neighbor* dan evaluasi yang menunjang pada penelitian yang akan dibuat.

BAB 3 METODOLOGI

Bab ketiga akan membahas terkait tipe penelitian, strategi penelitian, partisipan penelitian, lokasi penelitian, teknik pengumpulan data, teknik analisis data, peralatan pendukung penelitian, implementasi algoritme hingga jadwal penelitian.

BAB 4 PERANCANGAN

Bab keempat akan membahas terkait dengan analisis sistem dan perancangan untuk mengklasifikasikan berita ke dalam beberapa kategori yang ditentukan dengan menggunakan metode *Modified K-Nearest Neighbor* berdasarkan nilai *cosine similarity* yang dihasilkan.

BAB 5 IMPLEMENTASI

Bab kelima akan membahas terkait dengan implementasi yang dibangun berdasarkan hasil analisis sistem dan perancangan.

BAB 6 PENGUJIAN DAN ANALISIS

Bab keenam akan membahas terkait dengan hasil evaluasi dari hasil pengujian dan analisis sistem.

BAB 7 PENUTUP

Bab ketujuh akan membahas terkait dengan kesimpulan dan saran penelitian sehingga dapat dikembangkan pada penelitian selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan apa saja yang berkaitan dengan penelitian ini seperti kajian pustaka serta dasar teori yang akan menunjang dalam penelitian mengenai klasifikasi *hoax* pada berita kesehatan berbahasa Indonesia dengan metode *Modified K-Nearest Neighbor.*

2.1 Kajian Pustaka

Penelitian dibuat atas dasar studi kepustakaan pada penelitian sebelumnya dan yang berkaitan dengan Naïve Bayes dan *K-Nearest Neighbor* dalam klasifikasi teks. Studi kepustakaan disini ditujukan untuk dijadikan sumber kajian pustaka. Telah dilakukan penelitian untuk mengklasifikasikan kategori berita *hoax* atau fakta dengan menggunakan tiga algoritme yakni *Naïve Bayes*, *Support Vector Machine* dan *C4.5* untuk dibandingkan hasilnya (Rasywir & Purwarianti, 2015). Penelitian yang menggunakan data berupa 22 topik berita dan diklasifikasi menggunakan algoritme Naïve Bayes mampu menghasilkan nilai akurasi terbaik yakni sebesar 91.36%. Penghilangan *stopword* serta fitur tanpa *stemming* dapat memberikan akurasi yang lebih baik.

Keunggulan metode *Naïve Bayes* ini juga telah dibuktikan dalam penelitian mengenai klasifikasi teks homograf yang membandingkannya dengan metode *K-Nearest* Neighbor. Hasil penelitian menyebutkan bahwa dengan menggunakan algoritme *Naïve Bayes*, nilai *f-measure* lebih tinggi daripada *K-Nearest Neighbor* yakni dengan rata-rata 89%. Perolehan hasil yang rendah pada algoritme *K-Nearest Neighbor* dikarenakan sangat bergantung pada pemilihan nilai *k* (Anggono et al, 2009). Berbeda dengan kedua penelitian sebelumnya yang melakukan analisis terhadap berita *hoax* serta klasifikasi teks homograf dengan metode terbaik yakni Naïve Bayes. Penelitian kali ini akan menggunakan metode *Modified K-Nearest Neighbor* yang merujuk pada penelitian lain yang dilakukan oleh Palinoan dan Wijono (2014) mengenai klasifikasi dokumen berbahasa Jawa yang menggunakan metode *K-Nearest Neighbor*. Hasil akurasi yang diperoleh sangat tinggi yakni sebesar 95% dengan nilai *k* sebesar 4 dengan menggunakan evaluasi *cross validation* sejumlah 3 *fold*. Data pada penelitian tersebut menggunakan 40 dokumen percobaan dan 4 jenis kategori.

Pada penelitian ini akan menggunakan metode *Modified K-Nearest Neighbor,* langkah-langkah perhitungan algoritme tersebut adalah melakukan perhitungan *cosine similarity* dan perhitungan nilai validitas pada semua data latih lalu melakukan perhitungan *weight voting* terhadap data uji menggunakan validitas data dan *cosine distance* (Parvin et al, 2010). Berita dengan topik kesehatan akan digunakan sebagai data dan diharapkan dapat memberikan hasil yang lebih baik lagi dibandingkan dengan menggunakan algoritme *Naïve Bayes* (Rasywir & Purwarianti, 2015).

2.2 Berita

Pengertian berita menurut Kamus Besar Bahasa Indonesia (KBBI) adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Berita adalah laporan tercepat mengenai ide atau fakta terbaru yang benar, menarik dan penting bagi sebagian besar khalayak, melalui media berkala seperti surat kabar, radio, televisi, atau media internet (Sumadiria, 2005). Berita adalah laporan tentang peristiwa–peristiwa yang terjadi yang ingin diketahui oleh umum, dengan sifat aktual, terjadi di lingkungan pembaca, mengenai tokoh terkemuka, akibat peristiwa tersebut berpengaruh terhadap pembaca (Nasution dalam Abrar, 2005).

Sebuah berita harus mengandung unsur 5W+1H (*What, Who, When, Where, Why,* dan *How*) supaya pembaca dapat mengetahui lebih banyak dan detail tentang suatu kejadian. Penjelasan lebih rincinya adalah sebagai berikut (Cahya, 2012):

a. What

Suatu berita dikatakan baik jika memenuhi unsur *what*, yaitu berisi pernyataan yang dapat menjawab pertanyaan apa.

b. Who

Suatu berita dikatakan baik jika memenuhi unsur *who*, yaitu disertai keterangan tentang orang-orang yang terlibat dalam peristiwa.

c. When

Suatu berita dikatakan baik jika memenuhi unsur *when*, yaitu menyebutkan waktu kejadian peristiwa.

d. Where

Suatu berita dikatakan baik jika memenuhi unsur *where,* yaitu berisi deskripsi lengkap tentang tempat kejadian.

e. Why

Suatu berita dikatakan baik jika memenuhi unsur *why*, yaitu disertai alasan atau latar belakang terjadinya peristiwa.

f. How

Suatu berita dikatakan baik jika memenuhi unsur *how*, yaitu dapat dijelaskan proses kejadian suatu peristiwa dan akibat yang ditimbulkan.

2.3 *Hoax*

Hoax merupakan manipulasi berita yang sengaja dilakukan dan bertujuan untuk memberikan pengakuan atau pemahaman yang salah (Dahlan, 2017). Secara garis besar *hoax* adalah pemberitaan palsu untuk menipu orang lain demi keuntungan tertentu maupun popularitas. Salah satu contoh pemberitaan palsu yang paling umum adalah mengklaim sesuatu barang atau kejadian dengan suatu sebutan yang berbeda dengan barang/kejadian sejatinya. Berita *hoax* berbeda

dengan pertunjukkan sulap yang mana penipuan diutamakan agar penonton bisa mengapresiasi karya seni pesulap tersebut, berbeda halnya jika informasi *hoax* tersebar dan diterima oleh masyarakat yang mana bisa menyebabkan adu mulut yang berujung pada persekusi. Persekusi adalah penghakiman sendiri oleh elemen masyarakat tanpa ada campur tangan pihak yang berwajib.

Hoax kini menjadi perhatian serius pemerintah, contohnya saja rumor serbuan 10 juta pekerja China ke Indonesia. Memang benar adanya berita tersebut namun tidak benar-benar sampai 10 juta melainkan hanya sekitar 20-ribuan (Laoly, 2016). *Hoax* kebanyakan muncul dan tersebar di media sosial seperti Facebook, Whatsapp dan lainnya. Pelaku penyebar maupun pembuat berita *hoax* ini menerapkan pola *hit and run*, buka akun, lempar isu, tutup akun dan berulang-ulang begitu seterusnya yang menyebabkan susah untuk dianalisis (Setya, 2016).

Ketua Dewan Pers, Prasetyo (2017) menyebutkan bahwa ciri-ciri *hoax* adalah sebagai berikut:

a. Mengakibatkan kecemasan, kebencian, dan permusuhan.
b. Sumber berita tidak jelas. *Hoax* di media sosial biasanya pemberitaan media yang tidak terverifikasi, tidak berimbang, dan cenderung menyudutkan pihak tertentu.
c. Bermuatan fanatisme atas nama ideologi, judul, dan pengantarnya provokatif, memberikan penghukuman serta menyembunyikan fakta dan data.

2.4 *Preprocessing*

Dokumen-dokumen yang ada kebanyakan tidak memiliki struktur yang pasti sehingga informasi di dalamnya tidak bisa diekstrak secara langsung dan tidak semua kata mencerminkan makna yang terkandung dalam sebuah dokumen. Sebuah proses dengan nama *preprocessing* akan diperlukan untuk memilih kata yang akan digunakan sebagai indeks. Indeks ini adalah kata-kata yang mewakili dokumen yang nantinya digunakan untuk membuat pemodelan untuk *information retrieval* maupun aplikasi *text mining* lain. Singkatnya *preprocessing* adalah mengubah teks menjadi *term index* yang bertujuan untuk menghasilkan sebuah *set term index* yang bisa mewakili dokumen. Ada beberapa langkah dalam menerapkan *preprocessing* menurut (Fauzi, 2017) yakni:

1. *Parsing*

Proses awal disini dilakukan pemecahan struktur dokumen menjadi komponen yang terpisah, singkatnya adalah menentukan mana yang akan dijadikan satu dokumen. Contohnya buku dengan 500 halaman bisa dipecah menjadi 500 dokumen, yang mana per halaman dikelompokkan menjadi 1 dokumen.

2. *Lexical Analysis* atau Tokenisasi

Pada proses ini dilakukan penghilangan angka, tanda baca dan karakter selain huruf alfabet, karena karakter-karakter tersebut dianggap sebagai pemisah kata

(*delimiter*) dan tidak memiliki pengaruh terhadap pemrosesan teks. Pada tahapan ini juga dilakukan proses *case folding,* yakni perubahan pada awalan huruf semua kata menjadi awalan huruf kecil. Selanjutnya adalah tahapan *cleaning* yang berfungsi untuk menghilangkan informasi yang tidak berhubungan dengan dokumen contohnya *script, link, tag* HTML dan sebagainya.

3. *Stopword Removal* atau *Filtering*

Pada tahap ini dilakukan pemrosesan lanjut dari hasil tokenisasi. *Stoplist* atau *stopword* adalah kata-kata yang tidak penting yang dapat dibuang dengan pendekatan *bag-of-words*. Hasil dari *stoplist* adalah *wordlist* yang berisi kata penting.

4. Stemming

Stemming melakukan perubahan dari suatu kata menjadi kata dasar, yang mana setiap kata yang berimbuhan akan berubah menjadi kata dasar. Proses ini nantinya tidak digunakan karena berdasarkan penelitian sebelumnya, fitur tanpa *stemming* memberikan akurasi lebih baik, ini menunjukkan bahwa penciri berita *hoax* ditentukan secara leksikal (Rasywir & Purwarianti, 2015) . Leksikal adalah suatu makna yang nyata dalam kehidupan kita, jadi makna leksikal adalah arti sebenarnya yang dijelaskan oleh kata tersebut. Sebuah kata yang memiliki makna leksikal sudah jelas bahwa tanpa konteks pun memiliki makna langsung (Chaer, 2013).

2.5 Pembobotan Kata

Setelah indeks kata dari masing-masing dokumen telah diperoleh dari hasil *preprocessing*, selanjutnya dilakukan pembobotan kata untuk merubah menjadi bentuk numerik (Herwijayanti, 2018). Pembobotan kata disini tergantung pada jumlah kemunculan masing-masing *token* dalam dokumen. Sub bab 2.5.1 sampai 2.5.3 adalah proses perhitungan pembobotan kata (Fauzi, 2017).

2.5.1 *Term Frequency* (TF)

Merupakan banyaknya kemunculan *term*/*token*/kata t dalam dokumen d. Rumus *term frequency* akan dijelaskan pada Persamaan 2.1.

$$W_{tf_{t,d}} = \begin{cases} 1 + log_{10}\, tf_{t,d}, & if\ tf_{t,d} > 0 \\ 0, & otherwise \end{cases} \quad (2.1)$$

Keterangan:

$W_{tf_{t,d}}$ = Frekuensi kemunculan kata t dalam dokumen d

2.5.2 *Inverse Document Frequency* (IDF)

Merupakan banyaknya dokumen yang mengandung *term*/*token*/kata t. Rumus *inverse document frequency* akan dijelaskan pada Persamaan 2.2.

$$idf_t = log_{10}\ N/df_t \quad (2.2)$$

Keterangan:

df_t = Banyaknya dokumen yang memuat t

N = Jumlah total dokumen

2.5.3 TF.IDF *Weighting*

Bobot disini merupakan hasil perkalian dari $tf_{t,d}$ dan idf_t, Rumus *tf.idf weighting* akan dijelaskan pada Persamaan 2.3 dan normalisasinya pada Persamaan 2.4.

$$W_{t,d} = W_{tf_{t,d}} \, x \, idf_t \tag{2.3}$$

Normalisasi:

$$W_{t,d} = \frac{W_{t,d}}{\sqrt{\sum_{t=1}^{n} {W_{t,d}}^2}} \tag{2.4}$$

2.6 Cosine Similarity

Menurut Adikara et al. (2017) untuk membandingkan antar kata dalam beberapa dokumen dibutuhkan perhitungan yang tepat agar tingkat kemiripan bisa diperoleh. Dalam menghitung besarnya derajat kemiripan antara dokumen dan query dibutuhkan fungsi yang disebut *cosine similarity*. Nilai *cosine similarity* ditentukan berdasarkan perhitungan besarnya nilai fungsi *cosine* terhadap sudut yang dibentuk oleh dua vektor yakni pada penelitian ini adalah sebuah representasi dari dokumen-dokumen antar data latih. Rumus untuk menghitung tingkat kemiripan dokumen satu dengan dokumen lain akan dijelaskan pada Persamaan 2.5 dan normalisasinya pada Persamaan 2.6.

Tanpa normalisasi Wt,d:

$$CosSim(d_j, q) = \frac{\vec{d}_j \, . \, \vec{q}}{|\vec{d}_j| \, . \, |\vec{q}|} = \frac{\sum_{i=1}^{t}(W_{ij} \, . \, W_{iq})}{\sqrt{\sum_{i=1}^{t} {W_{ij}}^2 \, . \, \sum_{i=1}^{t} {W_{iq}}^2}} \tag{2.5}$$

Dengan normalisasi berdasarkan persamaan $W_{t,d}$ sebelumnya:

$$CosSim(d_j, q) = \vec{d}_j, \vec{q} = \sum_{i=1}^{t}(W_{ij} \, . \, W_{iq}) \tag{2.6}$$

Keterangan:

d_j = Data latih x

q = Tetangga data latih x

W_{ij} = Nilai pembobotan kata pada dokumen latih

W_{iq} = Nilai pembobotan kata pada tetangga dokumen latih

2.7 Modified K-Nearest Neighbor

Modified K-Nearest Neighbor adalah salah satu dari sekian algoritme yang digunakan untuk klasifikasi. Merupakan perkembangan dari algoritme *K-Nearest Neighbor* dan mempunyai cara kerja yang sama yakni mengelompokkan data baru

dengan *k* tetangga terdekat. Hal yang menjadi dasar dalam metode *Modified K-Nearest Neighbor* adalah melakukan perhitungan nilai validitas pada semua data latih. Selanjutnya, dilakukan perhitungan *weight voting* pada semua data uji menggunakan validitas data (Parvin et al, 2010).

2.7.1 Nilai Validitas

Tiap data latih yang akan dihitung dalam algoritme *Modified K-Nearest Neighbor* harus divalidasi dan tergantung pada setiap tetangga terdekatnya. Setelah dihitung maka hasil validitas tersebut digunakan sebagai informasi yang lebih mengenai data yang dihitung. Rumus untuk menghitung nilai validitas akan dijelaskan pada Persamaan 2.7.

$$Validity(x) = \frac{1}{H}\sum_{i=1}^{H} S(lbl(x), lbl(Ni(x))) \quad (2.7)$$

Keterangan:

$Validity$ = Validitas antar data latih

H = Jumlah tetangga terdekat

i = Nilai terbaik bernilai 1

$lbl(x)$ = Label kelas *x*

$lbl(Ni(x))$ = Label kelas titik terdekat dengan *x*

Fungsi S digunakan untuk menghitung kesamaan antara titik x dan data ke- i dari tetangga terdekat. Rumus untuk menghitung S akan dijelaskan pada Persamaan 2.8.

$$S(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \quad (2.8)$$

Keterangan:

S = *Similarity*

a = Kelas a pada data latih

b = Kelas selain a pada data latih

2.7.2 *Cosine Distance*

Dua titik yang satu pada data latih dan satunya lagi pada data uji akan dihitung jarak antara keduanya dijelaskan pada Persamaan 2.9 dan normalisasinya menggunakan Persamaan 2.10.

Cosine Distance:

$$1 - CosSim(d_j, q) = 1 - \frac{\vec{d}_j \,.\, \vec{q}}{|\vec{d}_j| \,.\, |\vec{q}|} = 1 - \frac{\sum_{i=1}^{t}(W_{ij} \,.\, W_{iq})}{\sqrt{\sum_{i=1}^{t} {W_{ij}}^2 \,.\, \sum_{i=1}^{t} {W_{iq}}^2}} \quad (2.9)$$

Normalisasi *Cosine Distance* berdasarkan persamaan $W_{t,d}$ sebelumnya:

$$1 - CosSim(d_j, q) = 1 - \vec{d}_j, \vec{q} = 1 - \sum_{i=1}^{t}(W_{ij} \,.\, W_{iq}) \quad (2.10)$$

Keterangan:

d_j = Data uji

q = Data latih x

W_{ij} = Nilai pembobotan kata pada dokumen uji

W_{iq} = Nilai pembobotan kata pada tetangga dokumen latih

2.7.3 *Weight Voting*

Langkah awal pada *weight voting* adalah *weight* masing-masing tetangga dihitung dengan menggunakan 1 / ($d_e + \alpha$), lalu dikalikan dengan validitas setiap data berdasarkan *cosine similarity.* Teknik ini mempunyai pengaruh lebih besar terhadap data yang memiliki nilai validitas lebih tinggi dan terdekat dengan data. Untuk setiap data yang mempunyai jarak dan *weight* yang bermasalah bisa diatasi dengan perkalian nilai validitas dengan jarak. Rumus untuk menghitung *weight voting* akan dijelaskan pada Persamaan 2.11.

$$W(i) = Validity(i)\ x\ \frac{1}{d_e + \alpha} \quad (2.11)$$

Keterangan:

W = Bobot antara data uji dengan data latih ke-i

i = Jumlah data latih

$Validity$ = Validitas data latih

d_e = Jarak data latih

α = *Regular smoothing* akan menggunakan nilai 0,5

2.8 Evaluasi

Evaluasi pada penelitian ini menggunakan evaluasi temu kembali tak berperingkat (Adikara et al, 2017), dalam menentukan perhitungan evaluasi akan merujuk pada Tabel 2.1.

Tabel 2.1 *Confusion Matrix*

Hasil Prediksi	**Hasil Aktual**	
	Hoax	**Fakta**
Hoax	TP	FP
Fakta	FN	TN

Sumber: Diadaptasi dari Nathania (2017)

Keterangan:

- *True Positive* (TP), yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.

- *True Negative* (TN), yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
- *False Positive* (FP), yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
- *False Negative* (FN), yaitu jumlah data positif namun terklasifikasi salah oleh sistem.

2.8.1 *Precision*

Precision adalah tingkat ketepatan antara informasi yang diminta oleh *user* dengan hasil jawaban yang diberikan oleh sistem. Dengan kata lain adalah perhitungan untuk menolak dokumen yang tidak relevan dalam dokumen. Rumus untuk menghitung *precision* akan dijelaskan pada Persamaan 2.12.

$$Precision = \frac{TP}{TP + FP} \tag{2.12}$$

2.8.2 *Recall*

Recall adalah tingkat jumlah banyak dan sedikitnya kesesuaian informasi yang didapatkan dari hasil percobaan berdasarkan sudut pandang kelas atau label yang digunakan. Singkatnya pada *recall* adalah perhitungan untuk menemukan semua dokumen yang relevan. Rumus untuk menghitung *recall* akan dijelaskan pada Persamaan 2.13.

$$Recall = \frac{TP}{TP + FN} \tag{2.13}$$

2.8.3 *F-Measure*

F-Measure adalah bobot *harmonic mean* pada *recall* dan *precision.* Jadi perhitungan antara *Precision* dan *Recall.* Rumus untuk menghitung *f-measure* akan dijelaskan pada Persamaan 2.14.

$$F = \frac{2 \times P \times R}{P+R} \tag{2.14}$$

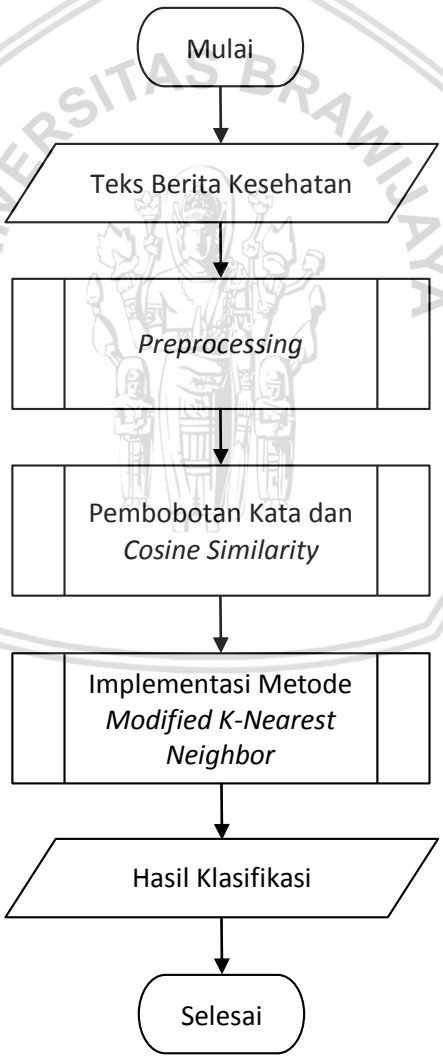
2.8.4 Akurasi

Akurasi adalah kesesuaian nilai hasil prediksi pengujian dengan nilai aktual (*ground truth)* yang dibandingkan. Rumus untuk menghitung akurasi akan dijelaskan pada Persamaan 2.15.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} * 100 \tag{2.15}$$

BAB 3 METODOLOGI

Dalam penelitian ini akan menggunakan metode *Modified K-Nearest Neighbor* dengan masukan sistem berupa dokumen teks berita kesehatan lalu akan diproses dengan *preprocessing* tanpa menggunakan *stemming*, dikarenakan arti kata dengan imbuhan mempunyai makna yang berbeda dengan kata dasarnya jadi hasilnya akan membuat kata kunci di dalam teks berita kesehatan tersebut akan berkurang dan dapat mengakibatkan hasil klasifikasi yang kurang tepat. pembobotan kata serta *cosine similarity* antar data latih. Selanjutnya mengklasifikasikan teks berita tersebut dengan menggunakan algoritme *Modified K-Nearest Neighbor*. Kemudian hasil akhir dari proses klasifikasi akan menghasilkan keluaran berupa kategori berita *hoax* atau fakta. Tahapan sistem secara umum digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Sistem

Metodologi yang dilakukan pada penelitian ini akan dijelaskan ke dalam beberapa subbab berikut:

3.1 Tipe Penelitian

Penelitian ini menggunakan jenis penelitian non-implementatif analitik. Prosesnya adalah menganalisis ulang penelitian yang sudah pernah dilakukan dengan menggunakan metode yang berbeda. Lalu pada akhirnya akan dilihat faktor-faktor apa saja yang mempengaruhi hasil penelitian.

3.2 Strategi Penelitian

Penelitian dimulai dari studi kepustakaan untuk mencari metode yang tepat dalam memecahkan suatu masalah agar bisa diketemukan solusinya. Referensi yang digunakan relevan dengan apa yang akan dibahas pada penelitian ini, yang bersumber dari buku, jurnal, *paper* maupun penelitian terkait. Studi kepustakaan yang diterapkan pada penelitian ini digunakan sebagai acuan dan landasan teori untuk dipelajari serta dianalisis lebih dalam terkait dengan permasalahan yang ada lalu dilanjutkan dengan perolehan dokumen berita kesehatan untuk diproses secara *preprocessing,* pembobotan kata, *cosine similarity* lalu dihitung dengan metode *Modified K-Nearest Neighbor* serta perhitungan evaluasi untuk tahap akhirnya, skenario yang akan dilakukan adalah sebagai berikut:

1. Masukan berupa data berita kesehatan yang digunakan sebagai data uji. Data di*preprocessing* lalu dilakukan pembobotan kata dan *cosine similarity* yang mana nanti hasilnya akan jadi acuan untuk metode *Modified K-Nearest Neighbor*.
2. Keluaran berupa hasil klasifikasi apakah berita termasuk ke dalam kategori *hoax* atau fakta.

3.3 Partisipan Penelitian

Partisipan dalam penelitian ini adalah dokter yang ditujukan untuk menjadi pakar dalam melabeli kategori kebenaran berita mengenai kesehatan dari sisi medis atau berdasarkan pengalaman dan ilmu pengetahuan dari dokter tersebut maupun rekan dokter lain.

3.4 Lokasi Penelitian

Lokasi pelaksanaan untuk pelabelan berita kesehatan yakni masing-masing di tempat praktik dan kediaman dari pakar tersebut. Lalu untuk penelitiannya dilakukan di Laboratorium Komputasi Cerdas Fakultas Ilmu Komputer Universitas Brawijaya, yang mana terdapat banyak sumber yang akan dijadikan kajian pustaka.

3.5 Teknik Pengumpulan Data

Bulan Maret hingga April tahun 2018 dilakukan pengumpulan data berupa teks berita yang diambil secara manual pada situs hoaxes.id, turnbackhoax.id dan vemale.com serta komunitas grup Facebook bernama Indonesian Hoaxes

Community. Dari beberapa jenis kategori berita yang akan diambil hanya berita kesehatan saja yakni sejumlah 170. Seluruh dari berita kesehatan tersebut nantinya akan dianalisis dahulu oleh pakar untuk melabelkan masing-masing berita menjadi 2 jenis yakni *hoax* dan fakta, berita yang tidak berhasil diklasifikasi akan langsung diberi pelabelan sesuai klarifikasi dari komunitas *hoax buster* dan portal berita. Hasil pelabelan tersebut nantinya akan digunakan sebagai data latih dan data uji.

3.6 Teknik Analisis Data

Beberapa aspek-aspek berikut akan diwujudkan dalam penelitian yakni sebagai berikut:

1. Pengguna sistem dapat mengubah atau menambah data uji.
2. Sistem dapat menampilkan hasil klasifikasi.

Setelah hasil klasifikasi berita diperoleh akan dilakukan analisis data agar hasil keluaran dari sistem bisa dihitung nilai ketepatannya dengan menggunakan *precision*, *recall, f-measure* dan akurasi. Nilai *k* dengan akurasi yang paling tinggi nantinya akan digunakan untuk memetakan hasil dari masing-masing data uji, apakah data aktual dengan hasil klasifikasinya sesuai atau tidak. Pengujian selanjutnya akan menggunakan *k-fold cross validation* yakni pemecahan seluruh data latih menjadi 10 bagian sama besar. Hasil perbandingan dengan metode *K-Nearest Neighbor* juga akan dilakukan guna melihat metode mana yang lebih baik.

3.7 Peralatan Pendukung Penelitian

Berikut adalah beberapa peralatan yang dibutuhkan untuk melakukan penelitian mengenai klasifikasi berita:

1. Perangkat keras yang digunakan spesifikasinya seperti berikut ini:
   a. Processor Intel® Core™ i5-4210U CPU @ 1.70GHz 2.40 GHz
   b. Kapasitas Memori (RAM) sebesar 4.00 GB
2. Perangkat lunak yang digunakan spesifikasinya seperti berikut ini:
   a. OS Windows 8.1 64 bit
   b. Python 3.7 64 bit

3.8 Implementasi Algoritme

Implementasi pada sistem menggunakan bahasa pemrograman python, dari pembuatan proses masukkan data hingga hasil keluaran berupa hasil klasifikasi mengenai berita kesehatan. Tahapannya dimulai dari implementasi pada *preprocessing,* pembobotan kata, *cosine similarity*, penerapan metode *Modified K-Nearest Neighbor* yang menggunakan bahasa pemrograman Python 3.7.

BAB 4 PERANCANGAN

Bab ini akan menjelasakan keseluruhan proses perancangan penelitian mulai dari deskripsi formulasi permasalahan, manualisasi, siklus algortime dan perancangan pengujian

4.1 Formulasi Permasalahan

Kasus yang diselesaikan pada penelitian ini adalah untuk mengklasifikasikan berita kesehatan agar nantinya bisa dikategorikan mana berita yang *hoax* dan mana berita yang fakta. Dengan jumlah persebaran berita yang saat ini sangat beragam dan banyak sekali, maka akan sulit dalam menentukan suatu berita termasuk *hoax* atau benar adanya. Terutama berita mengenai kesehatan yang banyak sekali disebar melalui grup di beberapa jejaring sosial bahkan situs yang *valid* pun tak luput dari persebaran berita kesehatan yang tidak dapat dipertanggungjawabkan keasliannya.

Klarifikasi berita kesehatan dengan berita dari sumber lainnya sangat sulit untuk dilakukan karena untuk pelabelan dari setiap berita yang tersebar harus diteliti dan didalami dahulu, oleh sebab itu penelitian ini akan mengklasifikasikan kategori berita kesehatan dengan data yang diperoleh secara manual dari *hoax buster* dan portal berita yang mana sejumlah 51 berita telah berhasil dilabeli oleh pakar berdasarkan pengelaman dunia kedokteran, 67 dilabeli oleh *hoax buster* berdasarkan hasil diskusi antar anggota *hoax buster* dan 52 berita sisanya dilabeli oleh portal berita berdasarkan klaim jurnalis. Data hasil klasifikasi dari ketiga sumber dapat dilihat pada Lampiran A. Sejumlah data yang akan digunakan untuk manualisasi dapat dilihat pada Tabel 4.1.

Tabel 4.1 Data Berita Kesehatan

No.	Artikel Berita Kesehatan	Kategori
1	Seperti diberitakan *Times* of *India*, Selasa (26/1/2016), penelitian yang diterbitkan dalam *New England Journal of Medicine* itu menunjukkan, orang-orang yang kerap menatap payudara wanita bisa hidup lebih lama. Studi ini mengklaim, 10 menit melirik bagian tubuh wanita setara dengan latihan gym 30 menit…...	*Hoax*
2	Nah, bicara soal tidur, tahukah Anda bahwa wanita membutuhkan waktu tidur yang lebih lama dibandingkan pria? Ya, sebuah studi yang belum lama ini dilakukan mengungkap bahwa otak wanita membutuhkan waktu istirahat yang lebih panjang…..	*Hoax*
3	Sebuah studi yang dilakukan di *University of Chicago* mengatakan, wanita dengan payudara besar lebih cerdas. Penelitian dilakukan dengan melibatkan 1.200 wanita…..	*Hoax*
4	Sebuah studi baru yang dikeluarkan Stanford University mengklaim bahwa "Satu kentut membakar sekitar 67 kalori.	*Hoax*

No.	Artikel Berita Kesehatan	Kategori
	Kentut 52 kali dalam satu hari dapat membakar 1 pon lemak." Selain itu, jumlah kalori yang dibakar oleh buang angin juga akan tergantung pada berapa lama buang angin itu berlangsung dan berapa banyak energi yang Anda gunakan untuk melakukannya…..	
5	…Rutin minum air putih yang cukup setiap hari juga bisa sangat membantu untuk proses BAB yang lebih lancar. Kalau kamu memiliki masalah sembelit atau konstipasi yang tak kunjung sembuh selama berhari-hari, sebaiknya langsung konsultasikan ke dokter untuk mendapatkan penanganan yang tepat, ya ladies.	Fakta
6	Nah *Ladies*, ternyata ada waktu minum air putih terbaik lho. Ini adalah di pagi hari, saat kamu bangun tidur. Minum satu gelas air putih, saat baru bangun akan membantu meningkatkan metabolisme tubuh. Selain itu ini akan berefek terhadap kesehatan pencernaan lho…..	Fakta
7	Sebagian orang menyarankan agar tidak mengonsumsi teh dicampur susu. Meski penelitian yang dilakukan pada tahun 1998 dan 2001 tidak menemukan bukti bahwa menambahkan susu bisa mengurangi manfaat kesehatan teh…..	Fakta

4.2 Manualisasi

Proses pengumpulan data yang telah dikumpulkan akan langsung dimasukkan ke dalam kodingan Python sebagai data latih dan data uji. Berdasarkan data dari Tabel 4.1 masing-masing contoh dari 2 kategori berita yang telah dilabeli dan dengan *term* yang sudah dikurangi akan digunakan untuk contoh perhitungan manualisasi.

4.2.1 *Preprocessing*

Tahap preprocessing digunakan untuk mendapatkan daftar kata atau *term* atau indeks kata yang mana akan mempermudah perhitungan pada tahapan selanjutnya yakni pembobotan kata. Proses awal dari *preprocessing* adalah *lexical analysis* atau tokenisasi yang digunakan untuk memecah kalimat menjadi kata per kata, setelah itu dilakukan proses pengubahan semua huruf menjadi huruf kecil dan penghilangan kata yang tidak berhubungan dengan dokumen seperti tanda baca, angka, *tag* HTML dan sebagainya. Dokumen yang akan dihitung pada manualisasi adalah berjumlah total 5 dengan jumah masing-masing 3 berita *hoax*, 2 berita fakta sebagai data latih dan 1 berita *hoax* serta 1 berita fakta sebagai data uji, hasil tokenisasi ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil Tokenisasi

Hasil Tokenisasi				Kategori
seperti	diberitakan	times	of	*Hoax*

Hasil Tokenisasi				Kategori
india	selasa	penelitian	yang	
diterbitkan	dalam	new	england	
journal	of	medicine	itu	
menunjukkan	orang-orang	yang	kerap	
menatap	payudara	wanita	bisa	
hidup	lebih	lama	studi	*Hoax*
ini	mengklaim	menit	melirik	
bagian	tubuh	wanita	setara	
dengan	latihan	gym	menit	
nah	bicara	soal	tidur	
tahukah	anda	bahwa	wanita	
membutuhkan	waktu	tidur	yang	
lebih	lama	dibandingkan	pria	
ya	sebuah	studi	yang	*Hoax*
belum	lama	ini	dilakukan	
mengungkap	bahwa	otak	wanita	
membutuhkan	waktu	istirahat	yang	
lebih		panjang		
sebuah	studi	yang	dilakukan	
di	university	of	chicago	
mengatakan	wanita	dengan	payudara	*Hoax*
besar	lebih	cerdas	penelitian	
dilakukan	dengan	melibatkan	wanita	
nah	ladies	ternyata	ada	
waktu	minum	air	putih	
terbaik	lho	ini	adalah	
di	pagi	hari	saat	
kamu	bangun	tidur	minum	
satu	gelas	air	putih	Fakta
saat	baru	bangun	akan	
membantu	meningkatkan	metabolisme	tubuh	
selain	itu	ini	akan	
berefek	terhadap	kesehatan	pencernaan	
lho				
sebagian	orang	menyarankan	agar	
tidak	mengonsumsi	the	dicampur	
susu	meski	penelitian	yang	
dilakukan	pada	tahun	dan	Fakta
tidak	menemukan	bukti	bahwa	
menambahkan	susu	bisa	mengurangi	
manfaat	kesehatan	teh		
studi	dikeluarkan	stanford	university	*Hoax*

Hasil Tokenisasi				Kategori
mengklaim	kentut	membakar	kalori	
kentut	kali	membakar	pon	
lemak	kalori	dibakar	buang	
angin	tergantung	buang	angin	*Hoax*
energi		melakukannya		
rutin	minum	air	putih	
yang	cukup	setiap	hari	
juga	bisa	sangat	membantu	
untuk	proses	bab	yang	
lebih	lancar	kalau	kamu	
memiliki	masalah	sembelit	atau	Fakta
konstipasi	yang	tak	kunjung	
sembuh	selama	berhari-hari	sebaiknya	
langsung	konsutasikan	ke	dokter	
untuk	mendapatkan	penanganan	yang	
tepat	ya		ladies	

Proses selanjutnya setelah tokenisasi adalah *filtering.* Proses *filtering* dilakukan untuk menghapus kata yang terdapat pada *stoplist* dan menyimpan kata-kata penting yang tidak termasuk di dalamnya. Hasil dari *filtering* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Hasil *Filtering*

Hasil *Filtering*				Kategori
diberitakan	times	of	india	
selasa	penelitian	diterbitkan	new	
england	journal	of	medicine	
orang-orang	kerap	menatap	payudara	*Hoax*
wanita	hidup	studi	mengklaim	
menit	melirik	tubuh	wanita	
setara	latihan	gym	menit	
bicara	tidur	tahukah	bahwa	
wanita	membutuhkan	tidur	dibandingkan	*Hoax*
pria	ya	studi	mengungkap	
otak	wanita	membutuhkan	isitrahat	
studi	university	of	chicago	
wanita	payudara	cerdas	penelitian	*Hoax*
melibatkan		wanita		
ladies	minum	air	putih	
terbaik	lho	pagi	bangun	
tidur	minum	gelas	air	Fakta
putih	bangun	membantu	meningkatkan	

Hasil *Filtering*				Kategori
metabolisme	tubuh	berefek	kesehatan	
pencernaan		lho		
orang	menyarankan	mengonsumsi	teh	Fakta
dicampur	susu	penelitian	menemukan	
bukti	susu	mengurangi	manfaat	
kesehatan		teh		
studi	dikeluarkan	stanford	university	*Hoax*
mengklaim	kentut	membakar	kalori	
kentut	kali	membakar	pon	
lemak	kalori	dibakar	buang	
angin	tergantung	buang	angin	
energi		melakukannya		
rutin	minum	air	putih	Fakta
membantu	proses	bab	lancar	
memiliki	sembelit	konstipasi	kunjung	
sembuh	berhari-hari	langsung	konsultasikan	
dokter	penanganan	ya	ladies	

4.2.2 Pembobotan Kata

Setelah tahapan *preprocessing* sudah dilakukan maka proses akan dilanjutkan dengan pembobotan kata. Contoh perhitungan manualisasinya menggunakan *term* air pada D4, langkah awalnya yakni menghitung TF menggunakan Persamaan 2.1.

$$W_{tf_{t,d}} = 1 + log_{10} 10 = 1 + 1 = 1{,}3010$$

Selanjutnya perhitungan df_t yakni banyaknya dokumen yang mengandung kata, data uji tidak dihitung. Dilanjutkan dengan perhitungan IDF menggunakan Persamaan 2.2.

$$idf_t = log_{10} 5/1 = 0{,}6990$$

Hasil perhitungan TF dan IDF keseluruhan akan ditunjukkan pada Tabel 4.4.

Tabel 4.4 Hasil Pembobotan Kata TF-IDF

INDEKS	*tf*							*df*	*idf*
	D1	D2	D3	D4	D5	D6	D7		
air	0	0	0	1,3010	0	0	1	1	0,6990
angin	0	0	0	0	0	1,3010	0	0	0
bab	0	0	0	0	0	0	1	0	0
bangun	0	0	0	1,3010	0	0	0	1	0,6990
berefek	0	0	0	1	0	0	0	1	0,6990
berhari-hari	0	0	0	0	0	0	1	0	0
bicara	0	1	0	0	0	0	0	1	0,6990

INDEKS	*tf*							*df*	*idf*
	D1	D2	D3	D4	D5	D6	D7		
buang	0	0	0	0	0	1,3010	0	0	0
bukti	0	0	0	0	1	0	0	1	0,6990
cerdas	0	0	1	0	0	0	0	1	0,6990
chicago	0	0	1	0	0	0	0	1	0,6990
dibakar	0	0	0	0	0	1	0	0	0
dibandingkan	0	1	0	0	0	0	0	1	0,6990
diberitakan	1	0	0	0	0	0	0	1	0,6990
dicampur	0	0	0	0	1	0	0	1	0,6990
dikeluarkan	0	0	0	0	0	1	0	0	0
diterbitkan	1	0	0	0	0	0	0	1	0,6990
dokter	0	0	0	0	0	0	1	0	0
energi	0	0	0	0	0	1	0	0	0
england	1	0	0	0	0	0	0	1	0,6990
gelas	0	0	0	1	0	0	0	1	0,6990
gym	1	0	0	0	0	0	0	1	0,6990
hidup	1	0	0	0	0	0	0	1	0,6990
india	1	0	0	0	0	0	0	1	0,6990
istirahat	0	1	0	0	0	0	0	1	0,6990
journal	1	0	0	0	0	0	0	1	0,6990
kali	0	0	0	0	0	1	0	0	0
kalori	0	0	0	0	0	1,3010	0	0	0
kentut	0	0	0	0	0	1,3010	0	0	0
kerap	1	0	0	0	0	0	0	1	0,6990
kesehatan	0	0	0	1	1	0	0	2	0,3979
konstipasi	0	0	0	0	0	0	1	0	0
konsultasikan	0	0	0	0	0	0	1	0	0
kunjung	0	0	0	0	0	0	1	0	0
ladies	0	0	0	1	0	0	1	1	0,6990
lancar	0	0	0	0	0	0	1	0	0
langsung	0	0	0	0	0	0	1	0	0
latihan	1	0	0	0	0	0	0	1	0,6990
lho	0	0	0	1,3010	0	0	0	1	0,6990
manfaat	0	0	0	0	1	0	0	1	0,6990
medicine	1	0	0	0	0	0	0	1	0,6990
melakukannya	0	0	0	0	0	1	0	0	0
melibatkan	0	0	1	0	0	0	0	1	0,6990
melirik	1	0	0	0	0	0	0	1	0,6990
membakar	0	0	0	0	0	1,3010	0	0	0
membantu	0	0	0	1	0	0	1	1	0,6990
membutuhkan	0	1,3010	0	0	0	0	0	1	0,6990
memiliki	0	0	0	0	0	0	1	0	0
menatap	1	0	0	0	0	0	0	1	0,6990

INDEKS	*tf*							*df*	*idf*
	D1	D2	D3	D4	D5	D6	D7		
menemukan	0	0	0	0	1	0	0	1	0,6990
mengklaim	1	0	0	0	0	1	0	1	0,6990
mengonsumsi	0	0	0	0	1	0	0	1	0,6990
mengungkap	0	1	0	0	0	0	0	1	0,6990
mengurangi	0	0	0	0	1	0	0	1	0,6990
meningkatkan	0	0	0	1	0	0	0	1	0,6990
menit	1,3010	0	0	0	0	0	0	1	0,6990
menyarankan	0	0	0	0	1	0	0	1	0,6990
metabolisme	0	0	0	1	0	0	0	1	0,6990
minum	0	0	0	1,3010	0	0	1	1	0,6990
new	1	0	0	0	0	0	0	1	0,6990
of	1,3010	0	1	0	0	0	0	2	0,3979
orang	0	0	0	0	1	0	0	1	0,6990
orang-orang	1	0	0	0	0	0	0	1	0,6990
otak	0	1	0	0	0	0	0	1	0,6990
pagi	0	0	0	1	0	0	0	1	0,6990
payudara	1	0	1	0	0	0	0	2	0,3979
penanganan	0	0	0	0	0	0	1	0	0
pencernaan	0	0	0	1	0	0	0	1	0,6990
penelitian	1	0	1	0	1	0	0	3	0,2219
pon	0	0	0	0	0	1	0	0	0
pria	0	1	0	0	0	0	0	1	0,6990
proses	0	0	0	0	0	0	1	0	0
putih	0	0	0	1,3010	0	0	1	1	0,6990
rutin	0	0	0	0	0	0	1	0	0
selasa	1	0	0	0	0	0	0	1	0,6990
sembelit	0	0	0	0	0	0	1	0	0
sembuh	0	0	0	0	0	0	1	0	0
setara	1	0	0	0	0	0	0	1	0,6990
stanford	0	0	0	0	0	1	0	0	0
studi	1	1	1	0	0	1	0	3	0,2219
susu	0	0	0	0	1,3010	0	0	1	0,6990
tahukah	0	1	0	0	0	0	0	1	0,6990
teh	0	0	0	0	1,3010	0	0	1	0,6990
terbaik	0	0	0	1	0	0	0	1	0,6990
tergantung	0	0	0	0	0	1	0	0	0
tidur	0	1,3010	0	1	0	0	0	2	0,3979
times	1	0	0	0	0	0	0	1	0,6990
tubuh	1	0	0	1	0	0	0	2	0,3979
university	0	0	1	0	0	1	0	1	0,6990
wanita	1,3010	1,3010	1,3010	0	0	0	0	3	0,2219

INDEKS	*tf*							*df*	*idf*
	D1	D2	D3	D4	D5	D6	D7		
ya	0	1	0	0	0	0	1	1	0,6990

Setelah TF dan IDF diperoleh maka langkah selanjutnya yakni melakukan perhitungan $W_{t,d}$ untuk *term* air pada D4 dengan Persamaan 2.3.

$$W_{t,d} = 2\ x\ 0{,}6990 = 0{,}9094$$

Hasil perhitungan $W_{t,d}$ keseluruhan akan ditunjukkan pada Tabel 4.5.

Tabel 4.5 Hasil Perhitungan $W_{t,d}$

INDEKS	$W_{t,d}$						
	D1	D2	D3	D4	D5	D6	D7
air	0	0	0	0,9094	0	0	0,6990
angin	0	0	0	0	0	0	0
bab	0	0	0	0	0	0	0
bangun	0	0	0	0,9094	0	0	0
berefek	0	0	0	0,6990	0	0	0
berhari-hari	0	0	0	0	0	0	0
bicara	0	0,6990	0	0	0	0	0
buang	0	0	0	0	0	0	0
bukti	0	0	0	0	0,6990	0	0
cerdas	0	0	0,6990	0	0	0	0
chicago	0	0	0,6990	0	0	0	0
dibakar	0	0	0	0	0	0	0
dibandingkan	0	0,6990	0	0	0	0	0
diberitakan	0,6990	0	0	0	0	0	0
dicampur	0	0	0	0	0,6990	0	0
dikeluarkan	0	0	0	0	0	0	0
diterbitkan	0,6990	0	0	0	0	0	0
dokter	0	0	0	0	0	0	0
energi	0	0	0	0	0	0	0
england	0,6990	0	0	0	0	0	0
gelas	0	0	0	0,6990	0	0	0
gym	0,6990	0	0	0	0	0	0
hidup	0,6990	0	0	0	0	0	0
india	0,6990	0	0	0	0	0	0
istirahat	0	0,6990	0	0	0	0	0
journal	0,6990	0	0	0	0	0	0
kali	0	0	0	0	0	0	0
kalori	0	0	0	0	0	0	0
kentut	0	0	0	0	0	0	0
kerap	0,6990	0	0	0	0	0	0

INDEKS	$W_{t,d}$						
	D1	D2	D3	D4	D5	D6	D7
kesehatan	0	0	0	0,3979	0,3979	0	0
konstipasi	0	0	0	0	0	0	0
konsultasikan	0	0	0	0	0	0	0
kunjung	0	0	0	0	0	0	0
ladies	0	0	0	0,6990	0	0	0,6990
lancar	0	0	0	0	0	0	0
langsung	0	0	0	0	0	0	0
latihan	0,6990	0	0	0	0	0	0
lho	0	0	0	0,9094	0	0	0
manfaat	0	0	0	0	0,6990	0	0
medicine	0,6990	0	0	0	0	0	0
melakukannya	0	0	0	0	0	0	0
melibatkan	0	0	0,6990	0	0	0	0
melirik	0,6990	0	0	0	0	0	0
membakar	0	0	0	0	0	0	0
membantu	0	0	0	0,6990	0	0	0,6990
membutuhkan	0	0,9094	0	0	0	0	0
memiliki	0	0	0	0	0	0	0
menatap	0,6990	0,6990	0	0	0	0	0
menemukan	0	0	0	0	0,6990	0,6990	0
mengklaim	0,6990	0,6990	0	0	0	0,6990	0,6990
mengonsumsi	0	0	0	0	0,6990	0,6990	0
mengungkap	0	0,6990	0,6990	0	0	0	0
mengurangi	0	0	0	0	0,6990	0,6990	0
meningkatkan	0	0	0	0,6990	0,6990	0	0
menit	0,9094	0	0	0	0	0	0
menyarankan	0	0	0	0	0,6990	0,6990	0
metabolisme	0	0	0	0,6990	0,6990	0	0
minum	0	0	0	0,9094	0	0	0,6990
new	0,6990	0	0	0	0	0	0
of	0,5177	0	0,3979	0	0	0	0
orang	0	0	0	0	0,6990	0	0
orang-orang	0,6990	0	0	0	0	0	0
otak	0	0,6990	0	0	0	0	0
pagi	0	0	0	0,6990	0	0	0
payudara	0,3979	0	0,3979	0	0	0	0
penanganan	0	0	0	0	0	0	0
pencernaan	0	0	0	0,6990	0	0	0
penelitian	0,2219	0	0,2219	0	0,2219	0	0
pon	0	0	0	0	0	0	0
pria	0	0,6990	0	0	0	0	0
proses	0	0	0	0	0	0	0

INDEKS	$W_{t,d}$						
	D1	D2	D3	D4	D5	D6	D7
putih	0	0	0	0,9094	0	0	0,6990
rutin	0	0	0	0	0	0	0
selasa	0,6990	0	0	0	0	0	0
sembelit	0	0	0	0	0	0	0
sembuh	0	0	0	0	0	0	0
setara	0,6990	0	0	0	0	0	0
stanford	0	0	0	0	0	0	0
studi	0,2219	0,2219	0,2219	0,2219	0	0,2219	0
susu	0	0	0	0	0,9094	0	0
tahukah	0	0,6990	0	0	0	0	0
teh	0	0	0	0	0,9094	0	0
terbaik	0	0	0	0,6990	0	0	0
tergantung	0	0	0	0	0	0	0
tidur	0	0,5177	0	0,3979	0	0	0
times	0,6990	0	0	0	0	0	0
tubuh	0,3979	0	0	0,3979	0	0	0
university	0	0	0,6990	0	0	0,6990	0
wanita	0,2886	0,2886	0,2886	0	0	0	0
ya	0	0,6990	0	0	0	0	0,6990
JUMLAH	10,3875	5.1360	2.4527	9.0070	5.7700	1.0263	2.9314

Hasil dari perhitungan $W_{t,d}$ untuk *term* air pada D4 di atas akan dinormalisasi dengan menggunakan Persamaan 2.4.

$$W_{t,d} = \frac{0,9094}{\sqrt{9,0070}} = \frac{0,9094}{3,0012} = 0,3030$$

Hasil perhitungan normalisasi $W_{t,d}$ keseluruhan akan ditunjukkan pada Tabel 4.6.

Tabel 4.6 Hasil Perhitungan Normalisasi $W_{t,d}$

INDEKS	Normalisasi $W_{t,d}$						
	D1	D2	D3	D4	D5	D6	D7
air	0	0	0	0,3030	0	0	0,4083
angin	0	0	0	0	0	0	0
bab	0	0	0	0	0	0	0
bangun	0	0	0	0,3030	0	0	0
berefek	0	0	0	0,2329	0	0	0
berhari-hari	0	0	0	0	0	0	0
bicara	0	0,3084	0	0	0	0	0
buang	0	0	0	0	0	0	0
bukti	0	0	0	0	0,2910	0	0
cerdas	0	0	0,4463	0	0	0	0

INDEKS	Normalisasi $W_{t,d}$						
	D1	D2	D3	D4	D5	D6	D7
chicago	0	0	0,4463	0	0	0	0
dibakar	0	0	0	0	0	0	0
dibandingkan	0	0,3084	0	0	0	0	0
diberitakan	0,2169	0	0	0	0	0	0
dicampur	0	0	0	0	0,2910	0	0
dikeluarkan	0	0	0	0	0	0	0
diterbitkan	0,2169	0	0	0	0	0	0
dokter	0	0	0	0	0	0	0
energi	0	0	0	0	0	0	0
england	0,2169	0	0	0	0	0	0
gelas	0	0	0	0,2329	0	0	0
gym	0,2169	0	0	0	0	0	0
hidup	0,2169	0	0	0	0	0	0
india	0,2169	0	0	0	0	0	0
istirahat	0	0,3084	0	0	0	0	0
journal	0,2169	0	0	0	0	0	0
kali	0	0	0	0	0	0	0
kalori	0	0	0	0	0	0	0
kentut	0	0	0	0	0	0	0
kerap	0,2169	0	0	0	0	0	0
kesehatan	0	0	0	0,1326	0,1657	0	0
konstipasi	0	0	0	0	0	0	0
konsultasikan	0	0	0	0	0	0	0
kunjung	0	0	0	0	0	0	0
ladies	0	0	0	0,2329	0	0	0,4083
lancar	0	0	0	0	0	0	0
langsung	0	0	0	0	0	0	0
latihan	0,2169	0	0	0	0	0	0
lho	0	0	0	0,3030	0	0	0
manfaat	0	0	0	0	0,2910	0	0
medicine	0,2169	0	0	0	0	0	0
melakukannya	0	0	0	0	0	0	0
melibatkan	0	0	0,4463	0	0	0	0
melirik	0,2169	0	0	0	0	0	0
membakar	0	0	0	0	0	0	0
membantu	0	0	0	0,2329	0	0	0,4083
membutuhkan	0	0,4013	0	0	0	0	0
memiliki	0	0	0	0	0	0	0
menatap	0,2169	0	0	0	0	0	0
menemukan	0	0	0	0	0,2910	0	0
mengklaim	0,2169	0	0	0	0	0,6899	0
mengonsumsi	0	0	0	0	0,2910	0	0

INDEKS	Normalisasi $W_{t,d}$						
	D1	D2	D3	D4	D5	D6	D7
mengungkap	0	0,3084	0	0	0	0	0
mengurangi	0	0	0	0	0,2910	0	0
meningkatkan	0	0	0	0,2329	0	0	0
menit	0,2822	0	0	0	0	0	0
menyarankan	0	0	0	0	0,2910	0	0
metabolisme	0	0	0	0,2329	0	0	0
minum	0	0	0	0,3030	0	0	0,4083
new	0,2169	0	0	0	0	0	0
of	0,1606	0	0,2541	0	0	0	0
orang	0	0	0	0	0,2910	0	0
orang-orang	0,2169	0	0	0	0	0	0
otak	0	0,3084	0	0	0	0	0
pagi	0	0	0	0,2329	0	0	0
payudara	0,1235	0	0,2541	0	0	0	0
penanganan	0	0	0	0	0	0	0
pencernaan	0	0	0	0,2329	0	0	0
penelitian	0,0688	0	0,1417	0	0,0924	0	0
pon	0	0	0	0	0	0	0
pria	0	0,3084	0	0	0	0	0
proses	0	0	0	0	0	0	0
putih	0	0	0	0,3030	0	0	0,4083
rutin	0	0	0	0	0	0	0
selasa	0,2169	0	0	0	0	0	0
sembelit	0	0	0	0	0	0	0
sembuh	0	0	0	0	0	0	0
setara	0,2169	0	0	0	0	0	0
stanford	0	0	0	0	0	0	0
studi	0,0688	0,0979	0,1417	0	0	0,2190	0
susu	0	0	0	0	0,3786	0	0
tahukah	0	0,3084	0	0	0	0	0
teh	0	0	0	0	0,3786	0	0
terbaik	0	0	0	0,2329	0	0	0
tergantung	0	0	0	0	0	0	0
tidur	0	0,2285	0	0,1326	0	0	0
times	0,2169	0	0	0	0	0	0
tubuh	0,1235	0	0	0,1326	0	0	0
university	0	0	0,4463	0	0	0,6899	0
wanita	0,0896	0,1274	0,1843	0	0	0	0
ya	0	0,3084	0	0	0	0	0,4083

4.2.3 *Cosine Similarity* Data Latih

Setelah proses pembobotan kata sudah dilakukan maka akan dilanjutkan dengan perhitungan tingkat kemiripan antar data latih, perhitungannya menggunakan Persamaan 2.6 karena sebelumnya telah melakukan perhitungan dengan normalisasi $W_{t,d}$. Berikut merupakan perhitungan antara D1 dengan D2:

$$CosSim(\mathrm{D1}, \mathrm{D2}) = (0{,}2169 * 0) + (0{,}06883 * 0{,}09789) + (0{,}0896 * 0{,}1274) + (0 * 0{,}3084) + \cdots = 0{,}018144$$

Hasil perhitungan *cosine similarity* keseluruhan akan ditunjukkan pada Tabel 4.7.

Tabel 4.7 Hasil *Cosine Similarity* Data Latih

	COSINE SIMILARITY				
	D1	**D2**	**D3**	**D4**	**D5**
D1	1	0,018144	0,108197	0,016372	0,006357
D2	0,018144	1	0,037339	0,030291	0
D3	0,108197	0,037339	1	0	0,013083
D4	0,016372	0,030291	0	1	0,021966
D5	0,006357	0	0,013083	0,021966	1

4.2.4 Proses Klasifikasi *Modified K-Nearest Neighbor*

Setelah diperoleh tingkat kemiripan antar data latih maka selanjutnya adalah mengurutkan nilainya berdasarkan dengan nilai *k* yang akan digunakan sebesar 3, jadi hanya 3 terbesar saja yang digunakan. Hasil pengurutan nilai *cosine similarity* ditunjukkan pada Tabel 4.8.

Tabel 4.8 Hasil Urutan *Cosine Similarity* Data Latih

	d(a,b)	***COSINE SIMILARITY***
D1	D1,D3	0,108196698
	D1,D2	0,018143831
	D1,D4	0,016371541
D2	D2,D3	0,037339104
	D2,D4	0,030291441
	D2,D1	0,018143831
D3	D3,D1	0,108196698
	D3,D2	0,037339104
	D3,D5	0,013082927
D4	D4,D2	0,030291441
	D4,D5	0,021966367
	D4,D1	0,016371541

	d(a,b)	*COSINE SIMILARITY*
D5	D5,D4	0,021966367
	D5,D3	0,013082927
	D5,D1	0,006357261

Setelah mendapatkan nilai kemiripan kata antar data latih selanjutnya dilakukan perhitungan dengan metode *Modified K-Nearest Neighbor* yakni dimulai dari perhitungan validitas data. Jika kategori dokumen sama maka bernilai *k*=1, jika kategori berbeda maka diisi *k*=0. Perhitungan validitas data menggunakan Persamaan 2.7 dengan contoh data dari D1 dengan D3, D2, D4.

$$Validity(\mathrm{D1}) = \frac{1}{3}(1 + 1 + 0) = 0{,}666667$$

Perhitungan validitas data keseluruhan ditunjukkan pada Tabel 4.9.

Tabel 4.9 Hasil Validitas Data

NO	*k*=1	*k*=2	*k*=3	*S(a,b)*	Validitas
1	1	1	0	2	0,666667
2	1	0	1	2	0,666667
3	1	1	0	2	0,666667
4	0	1	0	1	0,333333
5	1	0	0	1	0,333333

Langkah kedua yakni melakukan perhitungan *cosine similarity* data latih dengan data uji. Contoh perhitungannya yakni D1 dengan D6 yang mana menggunakan Persamaan 2.10.

$$\begin{aligned} Cosine\ Distance(\mathrm{D1}, \mathrm{D6}) &= 1 - CosSim(\mathrm{D1}, \mathrm{D6}) \\ &= 1 - (0{,}2169 * 0) + (0{,}2169 * 0{,}6899) \\ &\quad + (0{,}0688 * 0{,}2190) + (0 * 0{,}6899) + \cdots \\ &= 1 - 0{,}164702611 = 0{,}835297389 \end{aligned}$$

Perhitungan *cosine similarity* keseluruhan ditunjukkan pada Tabel 4.10.

Tabel 4.10 Hasil *Cosine Distance* Data Uji Dengan Data Latih

	COSINE DISTANCE
d(D6,D1)	0,835297389
d(D6,D2)	0,978563388
d(D6,D3)	0,661050196
d(D6,D4)	1
d(D6,D5)	1
d(D7,D1)	1
d(D7,D2)	0,874087266
d(D7,D3)	1

	COSINE DISTANCE
d(D7,D4)	0,438728504
d(D7,D5)	0

Setelah didapat nilai validitas data serta *cosine similarity* data uji dengan data latih maka akan dilanjutkan dengan perhitungan *weight voting* sesuai dengan Persamaan 2.9 dengan menggunakan contoh data D6 dengan D1.

$$W(\mathrm{D}(6,1)) = 0{,}666667\ x\ \frac{1}{0{,}835297389\ +\ 0{,}5} = 0{,}499264563$$

Perhitungan *weight voting* keseluruhan serta urutan berdasarkan nilai terbesarnya ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil Perhitungan *Weight Voting*

	WEIGHT VOTING	KATEGORI
d(D6,D3)	0,574192803	*HOAX*
d(D6,D1)	0,499264563	*HOAX*
d(D6,D2)	0,45088812	*HOAX*
d(D6,D4)	0,222222222	FAKTA
d(D6,D5)	0,222222222	FAKTA
d(D7,D4)	0,710180487	FAKTA
d(D7,D2)	0,485170544	*HOAX*
d(D7,D1)	0,444444444	*HOAX*
d(D7,D3)	0,444444444	*HOAX*
d(D7,D5)	0,444444444	FAKTA

Hasil akhir klasifikasi diambil berdasarkan nilai *k*=3 yang telah ditentukan lalu dijumlahkan sesuai kategori yang muncul. Dari Tabel 4.11 diperoleh 3 nilai terbesar yakni dengan jumlah total untuk D6 0,574192803 (*hoax*) + 0,499264563 (*hoax*) + 0,45088812 (*hoax*) = 1.524345486 (*hoax*) serta jumlah total untuk D7 0,710180487 (fakta) + 0,485170544 (*hoax*) + 0,485170544 (*hoax*) = 0,929614989 (*hoax*) dan 0,710180487 (fakta), maka kedua data uji menurut hasil perhitungan sistem memiliki kategori *hoax.*

4.2.5 Evaluasi

Ketepatan sistem dari hasil klasifikasi berita bisa dihitung berdasarkan hasil dari perhitungan 2 dokumen yang mana data latihnya sejumlah 3 dokumen berupa berita *hoax* dan 2 dokumen berupa berita fakta, dapat disimpulkan perhitungan evaluasinya pada Tabel 4.12 berikut:

Tabel 4.12 Hasil *Confusion Matrix*

Hasil Prediksi	Hasil Aktual	
	Hoax	**Fakta**
Hoax	1	1
Fakta	0	0

Perhitungan *precision* seperti yang sudah dijelaskan pada Persamaan 2.10.

$$Precision = \frac{1}{1+1} = 0{,}5$$

Perhitungan *recall* seperti yang sudah dijelaskan pada Persamaan 2.11.

$$Recall = \frac{1}{1+0} = 1$$

Perhitungan *f-measure* seperti yang sudah dijelaskan pada Persamaan 2.12.

$$F = \frac{2 \times 0{,}5 \times 1}{0{,}5+1} = \frac{1}{1{,}5} = 0.666667$$

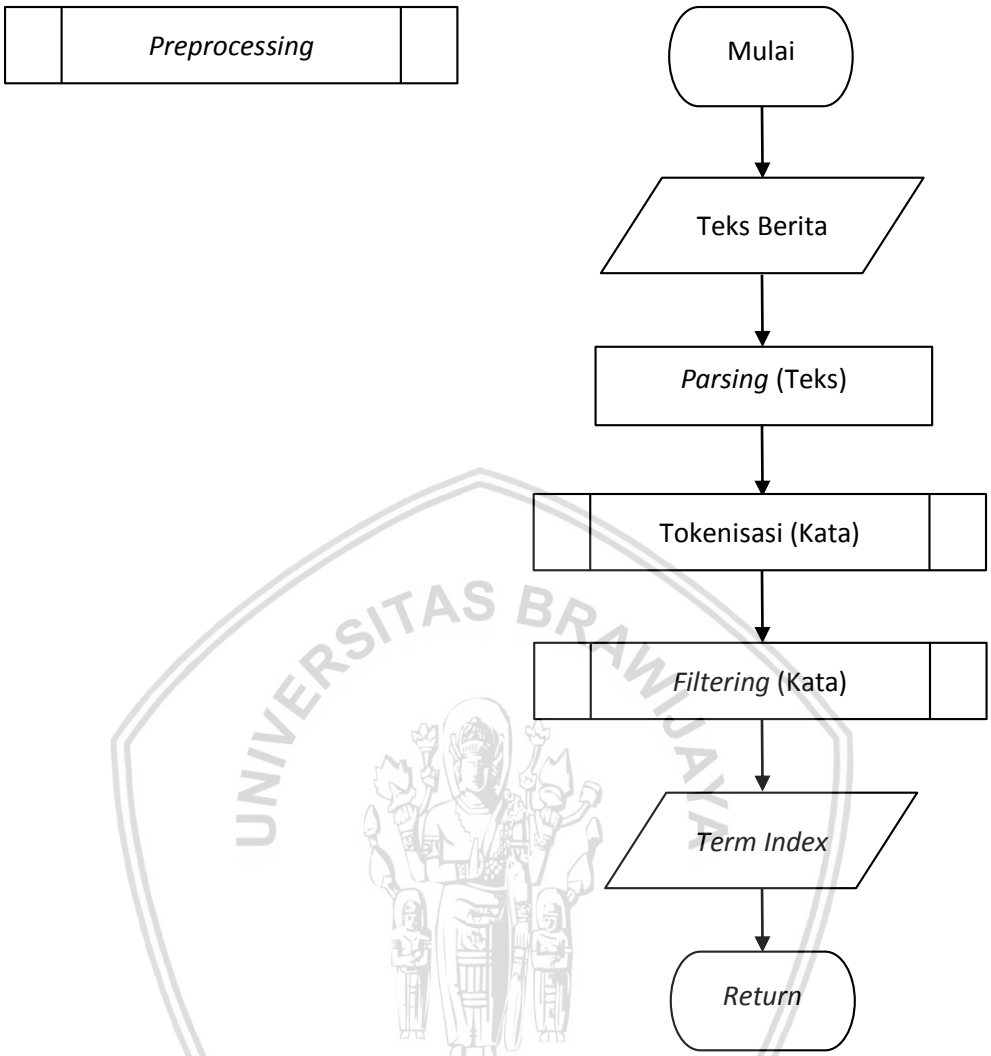
Perhitungan akurasi seperti yang sudah dijelaskan pada Persamaan 2.13.

$$Akurasi = \frac{1+0}{1+0+1+0} * 100 = \frac{1}{2} = 0{,}5 = 50\%$$

4.3 Siklus Algoritme

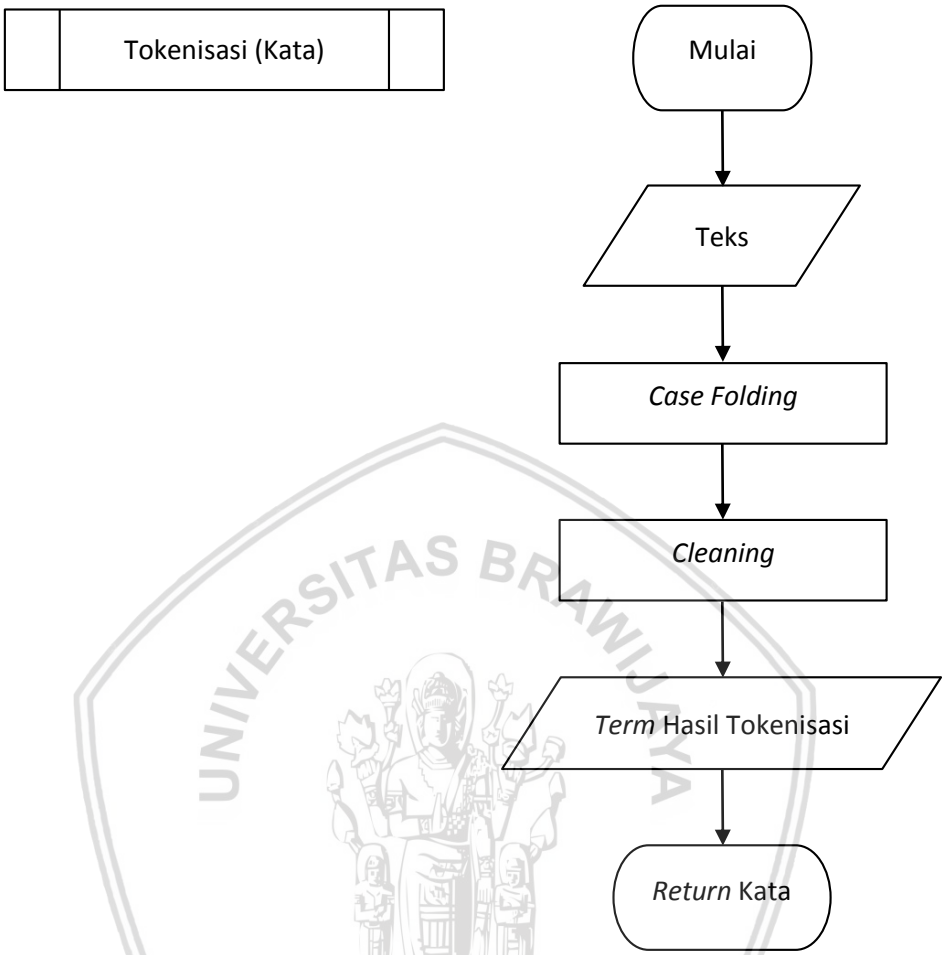
Pada sub bab ini akan dijelaskan gambaran setiap tahapan dari sistem klasifikasi berita dengan diagram alir yang mana mempunyai tujuan untuk menunjukkan langkah-langkah algoritme yang akan diterapkan pada penelitian ini. Penjelasan diagram alir sistem secara umum telah dijelaskan pada Gambar 3.1, untuk diagram alir tiap-tiap tahapan akan ditunjukkan pada sub bab ini.

Gambar 4.1 menjelaskan diagram alir dari tahapan *preprocessing* yang mana prosesnya untuk memperoleh *term index* dari suatu teks berita yang telah dilabeli oleh pakar, *hoax buster* dan portal berita dimulai dari *parsing* yakni untuk menentukan mana saja yang akan dijadikan 1 dokumen dengan kategori *hoax* atau kategori fakta, lalu *lexical analysis* atau tokenisasi untuk penghilangan karakter yang dianggap sebagai pemisah kata, *case folding* untuk merubah semua kata menjadi huruf kecil, *cleaning* untuk menghilangkan informasi yang tidak penting.



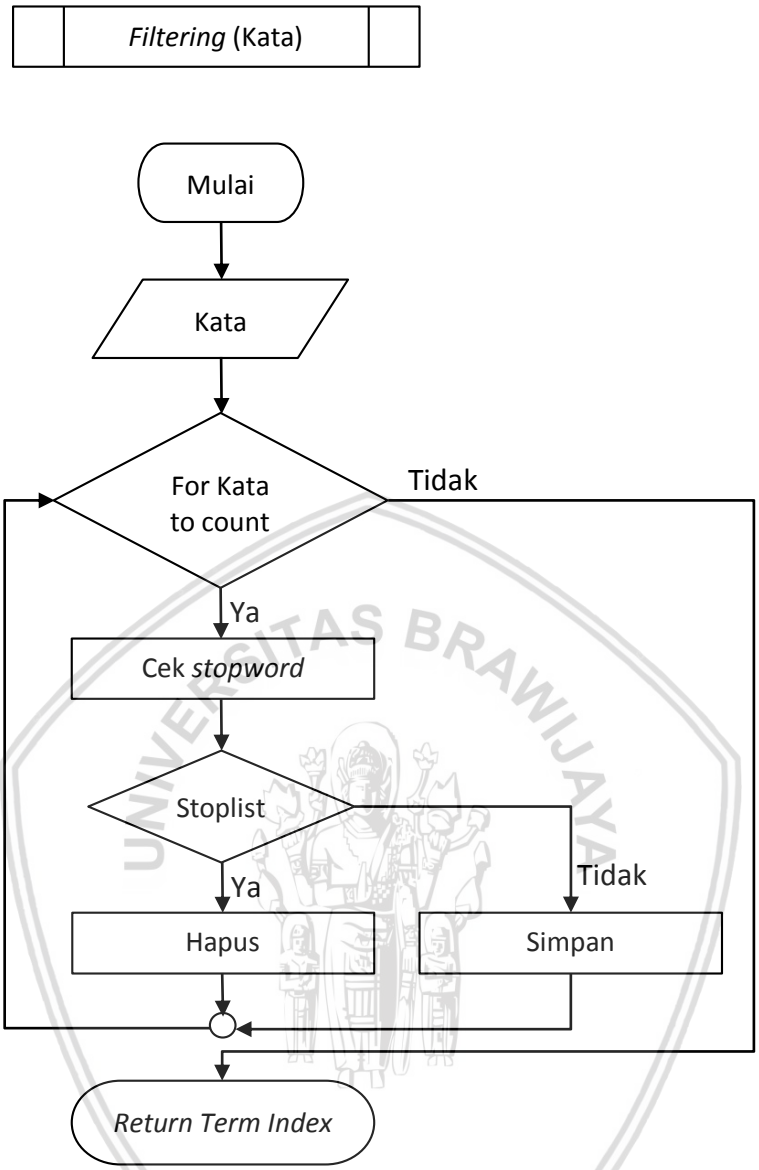
Gambar 4.1 Diagram Alir *Preprocessing*

Gambar 4.2 menjelaskan diagram alir dari tahapan tokenisasi yang mana prosesnya dimulai dari pemecahan teks berita berupa kalimat menjadi kata per kata, *case folding* untuk mengubah semua kata hasil dari tokenisasi dengan cara kata-kata yang telah diperoleh tersebut diubah menjadi kata-kata dengan huruf kecil dan *cleaning* untuk menghilangkan infomasi yang tidak berhubungan dengan dokumen seperti penghilangan angka, tanda baca, *link, script, tag* HTML dan karakter lainnya selain alfabet. Setelah itu didapatkan hasil berupa kata-kata yang sudah disetarakan atau yang telah dinormalisasi dan biasanya dikenal dengan sebutan *term.*



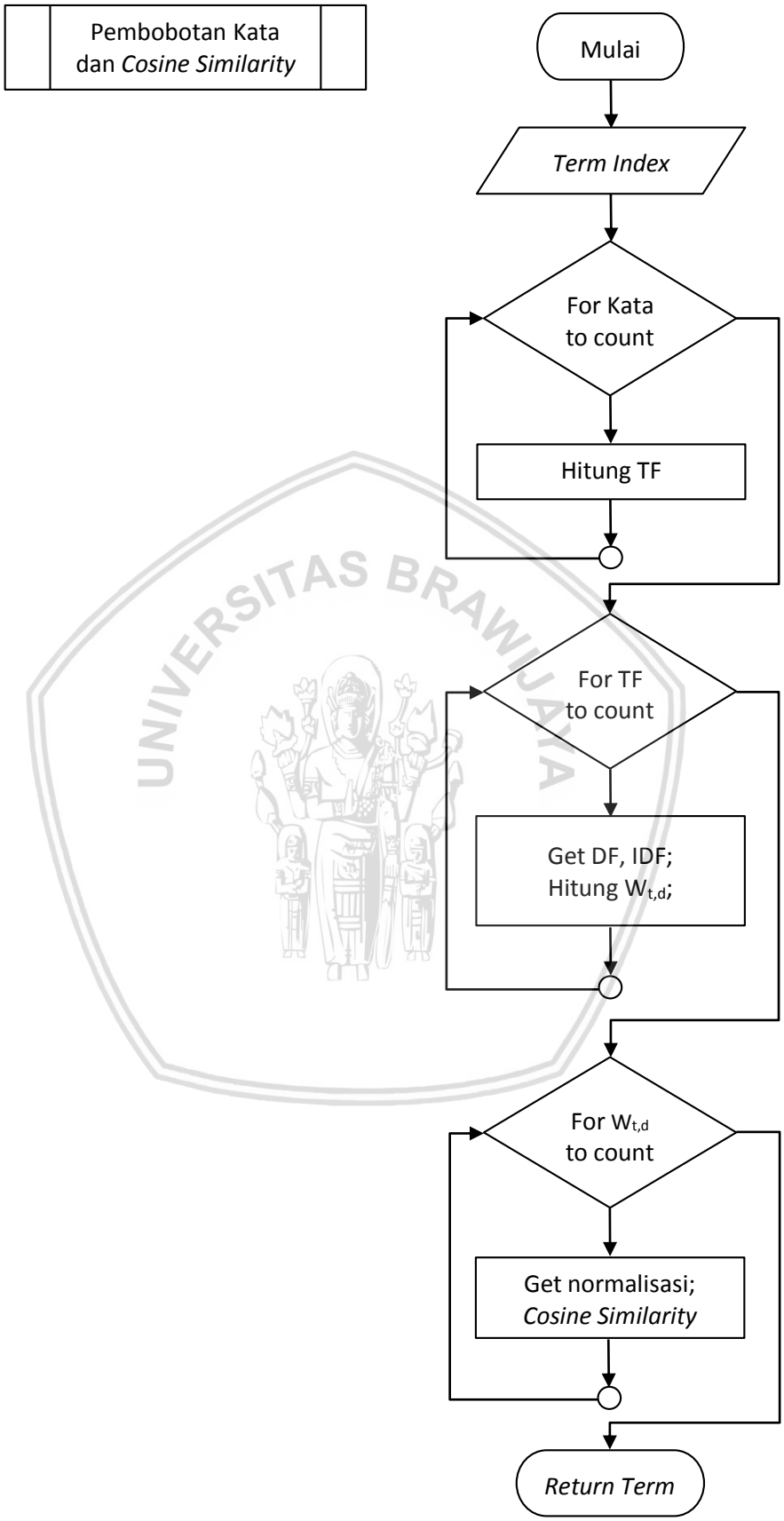
Gambar 4.2 Diagram Alir Tokenisasi

Gambar 4.3 menjelaskan diagram alir dari tahapan *filtering* yang bertujuan untuk menghapus atau menghilangkan kata yang terdapat di dalam *stoplist* yang dapat dibuang dengan pendekatan *bag of words*. Proses awalnya adalah mencocokkan setiap kata dengan *stoplist*, jika kata dalam dokumen tersebut ada di dalam *stoplist* maka kata tersebut akan dihilangkan dan menyimpan kata yang tidak termasuk di dalam *stoplist*. Perhitungan menggunakan fungsi *for* karena yang dihitung adalah semua kata. Hasil kata-kata atau *term index* yang telah dihasilkan dari tahapan *filtering* bisa disebut dengan istilah *wordlist* yakni kata penting untuk diproses ke tahapan selanjutnya yakni pembobotan kata dan *cosine similarity. Stoplist* yang akan digunakan didapat dari situs pak Putra Pandu Adikara, dapat dilihat pada Lampiran B.



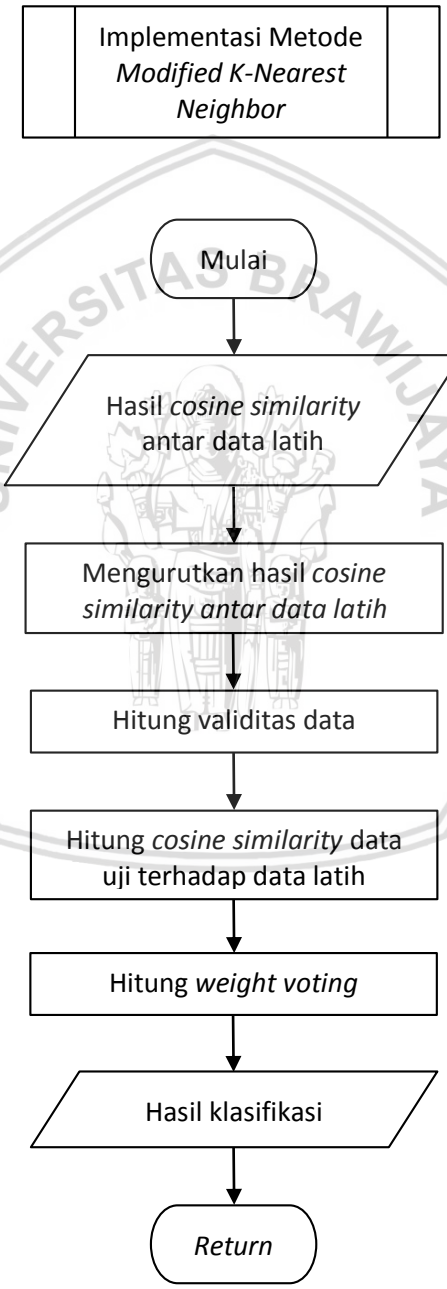
Gambar 4.3 Diagram Alir *Filtering*

Gambar 4.4 menjelaskan diagram alir dari tahapan pembobotan kata yang bertujuan untuk menghitung bobot tiap kata/*term* setelah proses *preprocessing* dilakukan. Proses perhitungan bobotnya yakni menghitung *term frequency atau* TF, *inverse document frequency* atau IDF, TF.IDF *weighting* serta hasil dari TF.IDF *weighting* atau $W_{t,d}$ nantinya akan dinormalisasi yang bertujuan untuk mempermudah perhitungan *cosine similarity* antar data latih. *Cosine similarity* adalah perhitungan untuk membandingkan tingkat kemiripan antar dokumen yang berbeda sehingga nantinya dapat diketahui derajat kemiripan yang dimiliki masing-masing dokumen, pada kasus ini digunakan untuk menghitung kemiripan antar data latih dan antara data latih dengan data uji.



Gambar 4.4 Diagram Alir Pembobotan Kata dan *Cosine Similarity*

Gambar 4.5 menjelaskan diagram alir dari tahapan klasifikasi menggunakan metode *Modified K-Nearest Neighbor* yang mana akan dilakukan setelah pembobotan kata dan perhitungan tingkat kemiripan antar data latih. Prosesnya yakni mengurutkan hasil perhitungan sesuai nilai *k,* menghitung validitas data, menghitung tingkat kemiripan antara dokumen uji terhadap dokumen latih yang sebelumnya data uji juga telah melalui proses *preprocessing* dan pembobotan kata. Tahapan akhir yakni perhitungan *weight voting*, perkalian antara validitas data dengan *cosine similarity* antara data latih dan data uji.



Gambar 4.5 Diagram Alir Klasifikasi Dengan Metode *Modified K-Nearest Neighbor*

4.4 Perancangan Pengujian

Perancangan ini digunakan untuk menghitung tingkat keakuratan dari sistem yang mengimplementasikan *Modified K-Nearest Neighbor.*

4.4.1 Pengujian *K-Values*

Pengujian akan menggunakan *k-values* yang berbeda-beda yakni menggunakan 2, 3, 4, 5, 7, 10, 15, 30, 75 dan 100 serta perhitungan *precision, recall, f-measure* dan akurasi dengan bantuan tabel *confusion matrix.* Tabel perancangan *k-values* ditunjukkan pada Tabel 4.13.

Tabel 4.13 Perancangan Tabel *K-Values*

k-Values	***Precision***	***Recall***	***F-Measure***	**Akurasi**

Setelah pengujian *k-values* dilakukan maka akan mendapatkan *k-values* terbaik, *k-values* tersebut akan digunakan untuk perancangan hasil klasifikasi data uji terhadap sistem yang ditunjukkan pada Tabel 4.14.

Tabel 4.14 Perancangan Tabel Hasil Klasifikasi Data Uji

K-Values	**Data Uji**	**Hasil**	
		Aktual	**Klasifikasi**
Akurasi			

4.4.2 Pengujian *K-Fold Cross Validation*

Pemilihan dokumen sebagai data uji bisa saja berpengaruh terhadap hasil klasifikasi dari sistem, maka dari itu dilakukan pengujian *k-fold cross validation*. Pengujian ini akan menggunakan 10-*fold* yang mana seluruh data akan dibagi menjadi 10 bagian sama rata. 1 bagian akan digunakan sebagai data uji dan 9 bagian sisanya akan digunakan sebagai data latih, iterasi dilakukan sebanyak 10 kali dengan pengambilan data uji yang bergeser setiap satu bagian. Tabel perancangan *k-fold cross validation* ditunjukkan pada Tabel 4.15.

Tabel 4.15 Perancangan Tabel *K-Fold Cross Validation*

K-Values	***Fold***	**Relevan**	**Tidak Relevan**	**Akurasi**
Rata-Rata Akurasi				

BAB 5 IMPLEMENTASI

Bab ini akan membahas mengenai implementasi dari hasil perancangan yang berisi batasan implementasi, implementasi dengan metode *Modified K-Nearest Neighbor* dan hasil pengujian implementasi.

5.1 Batasan Implementasi

Batasan implementasi digunakan untuk menjelaskan ruang lingkup dari implementasi sistem. Berikut merupakan beberapa batasan dari aplikasi klasifikasi berita kesehatan berbahasa Indonesia menggunakan metode *Modified K-Nearest Neighbor:*

1. Aplikasi klasifikasi berita kesehatan berbahasa Indonesia dirancang dan dijalankan menggunakan bahasa pemrograman Python 3.7.
2. Metode *Modified K-Nearest Neighbor* digunakan untuk menyelesaikan masalah.
3. Data yang digunakan berupa berita kategori kesehatan.
4. Pelabelan berita didapat dari hasil klarifikasi dari pakar sejumlah 18 berita fakta dan 33 berita *hoax*, hasil klarifikasi tim *hoax buster* sejumlah 16 berita fakta dan 51 berita *hoax* dan pengambilan langsung dari portal berita sejumlah 51 berita fakta 1 berita *hoax.*
5. Penentuan klasifikasi berdasarkan frekuensi kemunculan kata pada dokumen.
6. Tidak ada kata khusus dalam mengklasifikasikan berita jadi semua kata akan digunakan.
7. *Stemming* tidak akan digunakan pada proses *preprocessing.*

5.2 Implementasi

Sistem klasifikasi *hoax* pada berita kesehatan ini terdiri dari beberapa proses yakni dari *preprocessing*, pembobotan kata, *cosine similarity* data latih dan klasifikasi yang menggunakan metode *Modified K-Nearest Neighbor*. Teks berita kesehatan akan digunakan untuk masukkan lalu akan diolah sehingga menghasilkan keluaran berupa hasil kategori *hoax* atau fakta.

5.2.1 *Preprocessing*

Preprocessing terdiri dari dua tahapan yang pertama tokenisasi yakni merubah teks kalimat menjadi kata per kata dan yang kedua *filtering* yakni membuang kata yang termasuk ke dalam *stoplist* dan menyimpan kata yang tidak termasuk ke dalamnya ditunjukkan pada Kode Program 5.1.

```
1  def tokenisasi(list_berita):
2      list_kata = []
3      for berita in list_berita:
4          berita = berita.lower()
5          daftar_kata = re.findall(regex_kata, berita)
```

```
6           for kata in daftar_kata:
7               if (kata not in list_kata) and (kata not in stopword):
8                   list_kata.append(kata)
9       return list_kata
10
11  list_kata = tokenisasi(list_berita)
12  print("\n#### KATA ####")
13  for kata in list_kata:
14      print(kata)
```

Kode Program 5.1 Implementasi *Preprocessing*

Pembahasan Kode Program 5.1:

- Baris 1 : Fungsi *preprocessing* untuk menyimpan kata-kata ke dalam *array*.
- Baris 2 : Deklarasi *array* untuk menyimpan hasil perhitungan *preprocessing.*
- Baris 3-5 : Melakukan perhitungan tokenisasi.
- Baris 6-8 : Melakukan perhitungan *filtering*.
- Baris 9 : Menghentikan dan mengembalikan hasil perhitungan dari fungsi yang telah dijalankan.
- Baris 11-14 : Menampikan hasil dari perhitungan *preprocessing.*

5.2.2 *Pembobotan Kata*

Pembobotan kata menggunakan tf-idf, perhitungan awal mencari nilai masing-masing tf dan df lalu keduanya dikalikan. Setelah itu pencarian nilai idf lalu dilakukan perhitungan wtd. Normalisasi wtd adalah langkah terakhir yang digunakan untuk mempermudah perhitungan *cosine similarity.*

```
1   def pembobotanKata(list_kata):
2       list_df = []
3       list_tf = []
4       for kata in list_kata:
5           tf = []
6           df = 0
7           index = 0
8           for berita in list_berita:
9               total = 0
10              berita = berita.lower()
11              daftar_kata = re.findall(regex_kata, berita)
12              for item in daftar_kata:
13                  if kata == item:
14                      total += 1
15              tf.append(total)
16              if total > 0 and index < train:
17                  df += 1
18              index += 1
19          list_tf.append(tf)
20          list_df.append(df)
21      return list_tf, list_df
22
23  def hitungWtf(list_tf):
24      list_wtf = list_tf
25      for i in range(len(list_wtf)):
26          for j in range(len(list_wtf[i])):
27              if list_wtf[i][j] > 0:
```

```
                list_wtf[i][j]=1+ math.log10(list_wtf[i][j])
    return list_wtf

list_wtf = hitungWtf(list_tf)
print("\n#### TF ####")
for i in range(len(list_wtf)):
    print("%-15s%-15s" % (list_kata[i], list_wtf[i]))

def hitungIdf(list_df):
    list_idf = []
    for df in list_df:
        if df != 0:
            list_idf.append(math.log10(train/df))
        else:
            list_idf.append(0)
    return list_idf

list_idf = hitungIdf(list_df)
print("\n#### IDF ####")
for i in range(len(list_idf)):
    print("%-15s%-15s" % (list_kata[i], list_idf[i]))

def hitungWtd(list_wtf, list_idf):
    list_wtd = []
    for i in range(len(list_wtf)):
        temp = []
        for j in range(len(list_wtf[i])):
            temp.append(list_wtf[i][j]*list_idf[i])
        list_wtd.append(temp)
    return list_wtd

list_wtd = hitungWtd(list_wtf, list_idf)
print("\n#### WTD ####")
for i in range(len(list_kata)):
    print("%-15s%-15s" % (list_kata[i], list_wtd[i]))

def normalisasiWtd(list_wtd):
    list_wtd_kuadrat = []
    for i in range(len(list_wtd)):
        temp = []
        for j in range(len(list_wtd[i])):
            temp.append(list_wtd[i][j]**2)
        list_wtd_kuadrat.append(temp)
    jumlah = []
    for i in list_berita:
        jumlah.append(0)

    for wtd_kuadrat in list_wtd_kuadrat:
        for i in range(len(wtd_kuadrat)):
            jumlah[i] += wtd_kuadrat[i]
    jumlah = [math.sqrt(jml) for jml in jumlah]
    list_normalisasi_wtd = []
    for wtd in list_wtd:
        temp = []
        for i in range(len(wtd)):
            temp.append(wtd[i]/jumlah[i])
        list_normalisasi_wtd.append(temp)
    return list_normalisasi_wtd, jumlah
list_normalisasi_wtd, jumlah = normalisasiWtd(list_wtd)
print("\n#### NORMALISASI WTD ####")
for i in range(len(list_kata)):
    print("%-15s%-15s" % (list_kata[i], list_normalisasi_wtd[i])
```

Kode Program 5.2 Implementasi Pembobotan Kata

Pembahasan Kode Program 5.2:

- Baris 1 : Fungsi pembobotan kata untuk menghitung masing-masing kata berdasarkan kata yang telah tersimpan di dalam *list* kata.
- Baris 2-3 : Deklarasi *array* untuk menyimpan hasil perhitungan *document frequency* dan *term frequency.*
- Baris 4-11 : Melakukan perhitungan jumlah kata dengan melakukan *preprocessing* lagi.
- Baris 12-15, 19 : Melakukan perhitungan jumlah tf.
- Baris 16-18, 20 : Melakukan perhitungan jumlah df.
- Baris 21 : Menghentikan dan mengembalikan hasil perhitungan dari fungsi yang telah dijalankan.
- Baris 23 : Fungsi perhitungan tf tiap kata.
- Baris 24-28 : Melakukan perhitungan tf.
- Baris 29 : Menghentikan dan mengembalikan hasil perhitungan dari fungsi yang telah dijalankan.
- Baris 31-34 : Menampikan hasil dari perhitungan tf.
- Baris 36 : Fungsi perhitungan idf tiap kata.
- Baris 37 : Deklarasi *array* untuk menyimpan hasil perhitungan *inverse document frequency.*
- Baris 38-42 : Melakukan perhitungan idf tiap kata.
- Baris 43 : Menghentikan dan mengembalikan hasil perhitungan dari fungsi yang telah dijalankan.
- Baris 45-48 : Menampikan hasil dari perhitungan idf.
- Baris 50 : Fungsi perhitungan wtd tiap kata.
- Baris 51 : Deklarasi *array* untuk menyimpan hasil perhitungan *wtd.*
- Baris 52-56 : Melakukan perhitungan wtd tiap kata.
- Baris 57 : Menghentikan dan mengembalikan hasil perhitungan dari fungsi yang telah dijalankan.
- Baris 59-62 : Menampikan hasil dari perhitungan wtd.
- Baris 64 : Fungsi perhitungan normalisasi wtd tiap kata.
- Baris 65 : Deklarasi *array* untuk menyimpan hasil perhitungan jumlah *kuadrat* wtd.
- Baris 66-73 : Melakukan perhitungan jumlah kuadrat wtd tiap kata.

- Baris 75-78 : Melakukan perhitungan jumlah kuadrat wtd tiap kata.
- Baris 79 : Deklarasi *array* untuk menyimpan hasil perhitungan normalisasi *wtd.*
- Baris 80-84 : Melakukan perhitungan normalisasi wtd tiap kata.
- Baris 85 : Menghentikan dan mengembalikan hasil perhitungan dari fungsi yang telah dijalankan.
- Baris 86-89 : Menampikan hasil dari perhitungan normalisasi wtd.

5.2.3 *Cosine Similarity*

Nilai kemiripan antar dokumen latih dihitung dan diurutkan berdasarkan nilai *k* yang telah ditentukan.

```
1  list_cossin = []
2  for i in range(len(list_berita)):
3      temp1 = []
4      for j in range(len(list_berita)):
5          total = 0
6          for k in range(len(list_normalisasi_wtd)):
7              total+=
8  list_normalisasi_wtd[k][i]*list_normalisasi_wtd[k][j]
9          temp1.append(total)
10     list_cossin.append(temp1)
11
12 print("\n#### COSINE SIMILARITY ####")
13 for i in range(len(list_cossin)):
14     print("D" + str(i+1), list_cossin[i])
15
16 sort_cossin = []
17 for i in range(train):
18     temp = []
19     for j in range(train):
20         if i != j:
21             arr = []
22             arr.append(i)
23             arr.append(j)
24             arr.append(list_cossin[i][j])
25             temp.append(arr)
26     temp.sort(key=itemgetter(2), reverse=True)
27     temp2 = []
28     for j in range(nilai_k):
29         temp2.append(temp[j])
30     sort_cossin.append(temp2)
31
32 print("\n#### URUTAN COSINE SIMILARITY (k) ####")
33 for item in sort_cossin:
34     print(item)
```

Kode Program 5.3 Implementasi *Cosine Similarity*

Pembahasan Kode Program 5.3:

- Baris 1 : Deklarasi *array* untuk menyimpan hasil perhitungan *cosine similarity.*
- Baris 2-10 : Melakukan perhitungan *cosine similarity* tiap data latih berdasarkan nilai *k*.

- Baris 12-14 : Menampikan hasil dari perhitungan *cosine similarity.*
- Baris 16 : Deklarasi *array* untuk menyimpan hasil pengurutan *cosine similarity.*
- Baris 17-30 : Melakukan pengurutan *cosine similarity* sesuai nilai *k*.
- Baris 32-34 : Menampikan hasil dari perhitungan *cosine similarity*.

5.2.4 Proses Klasifikasi *Modified K-Nearest Neighbor*

Diawali dari perhitungan validitas data yakni apakah tiap-tiap data latih mempunyai kategori yang sama atau tidak. Proses selanjutnya menghitung *cosine distance* yakni perhitungan nilai 1 dikurangi *cosine similarity* data latih dengan data uji. Langkah terakhir yakni perhitungan *weight voting* yang berupa perkalian antara validitas data dibagi dengan *cosine distance* yang sebelumnya telah ditambah nilai *regular smoothing* sebesar 0,5. Setelah semua proses selesai maka hasil akhir dari klasifikasi menggunakan *Modified K-Nearest Neighbor* didapatkan.

```
1  validitas = []
2  for i in sort_cossin:
3      total = 0
4      for j in range(nilai_k):
5          if list_hoax[j[0]] == list_hoax[j[1]]:
6              total += 1
7      total = total / nilai_k
8      validitas.append(total)
9
10 print("\n#### VALIDITAS DATA ####")
11 for i in range(len(validitas)):
12     print("D" + str(i+1) + " :", validitas[i])
13
14 cosine_distance = []
15 for i in range(train, len(list_cossin)):
16     temp = []
17     for j in range(train):
18         temp.append(1-list_cossin[i][j])
19     cosine_distance.append(temp)
20
21 print("\n#### COSINE DISTANCE ####")
22 for item in cosine_distance:
23     print(item)
24
25 weight_voting = []
26 for i in range(len(list_berita)-train):
27     temp = []
28     for j in range(len(cosine_distance[i])):
29         weight = validitas[j]/(cosine_distance[i][j]+0.5)
30         temp2 = []
31         temp2.append(j)
32         temp2.append(weight)
33         temp.append(temp2)
34     temp.sort(key=itemgetter(1), reverse=True)
35     weight_voting.append(temp)
36
37 for i in range(len(weight_voting)):
38     print("\n#### WEIGHT VOTING Data Uji",i+train+1, "####")
39     for j in range(nilai_k):
40         kls = ""
41         if list_hoax[weight_voting[i][j][0]] == 1:
42             kls = "Hoax"
```

```
43          else:
44              kls = "Fakta"
45          print(weight_votingr[i][j], " = ", kls)
46
47  klasifikasi = []
48  for i in weight_voting:
49      totalHoax = 0
50      totalFakta = 0
51      temp = []
52      for j in range(nilai_k):
53          if list_hoax[i[j][0]] == 1:
54              totalHoax += i[j][1]
55          else:
56              totalFakta += i[j][1]
57      if totalHoax > totalFakta:
58          temp.append("Hoax")
59      else:
60          temp.append("Fakta")
61      temp.append(totalHoax)
62      temp.append(totalFakta)
63      klasifikasi.append(temp)
64
65  print("\n#### HASIL KLASIFIKASI ####")
66  for i in range(len(klasifikasi)):
67      asli = ""
68      if list_hoax[i+train] == 1:
69          asli = "Hoax"
70      else:
71          asli = "Fakta"
72      print("Data Uji :", (i+train+1), "-> Total Hoax :",
73  klasifikasi[i][1], "| Total Fakta :", klasifikasi[i][2])
74      print("                Data Aktual :", asli, "- > Hasil
75  Klasifikasi :", klasifikasi[i][0])
```

Kode Program 5.4 Implementasi *Modified K-Nearest Neighbor*

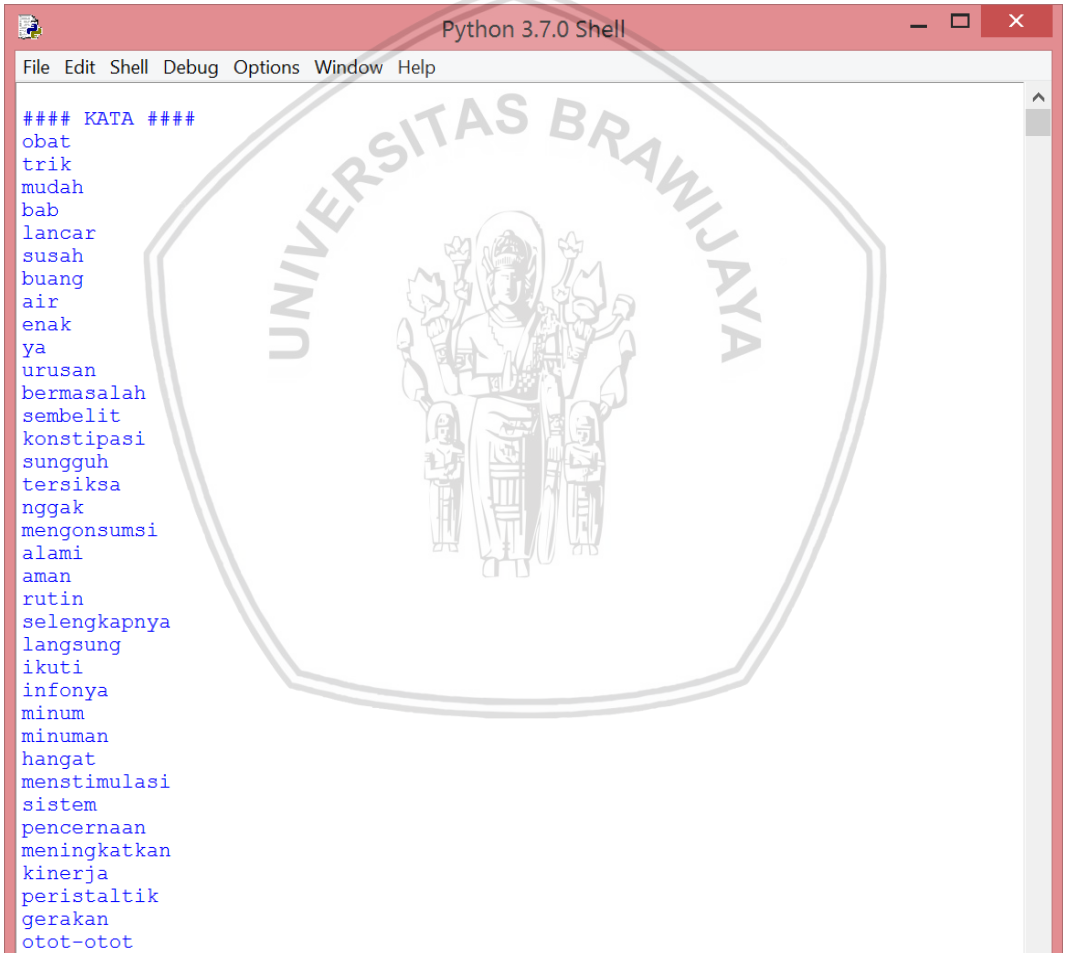
Pembahasan Kode Program 5.4:

- Baris 1 : Deklarasi *array* untuk menyimpan hasil perhitungan validitas data.
- Baris 2-8 : Melakukan perhitungan validitas data berdasarkan hasil pengurutan *cosine similarity.*
- Baris 10-12 : Menampikan hasil dari perhitungan validitas data.
- Baris 14 : Deklarasi *array* untuk menyimpan hasil perhitungan *cosine distance.*
- Baris 15-19 : Melakukan perhitungan *cosine distance* data latih dengan data uji.
- Baris 21-23 : Menampikan hasil dari perhitungan *cosine distance*.
- Baris 25 : Deklarasi *array* untuk menyimpan hasil perhitungan *weight voting.*
- Baris 26-35 : Melakukan perhitungan *weight voting* berdasarkan hasil validitas data dengan *cosine distance.*
- Baris 37-45 : Menampikan hasil dari perhitungan *weight voting.*

- Baris 47 : Deklarasi *array* untuk menyimpan hasil klasifikasi.
- Baris 48-63 : Melakukan perhitungan klasifikasi berdasarkan *weight voting* terbesar dari dua kategori.
- Baris 65-75 : Menampikan hasil dari perhitungan klasifikasi.

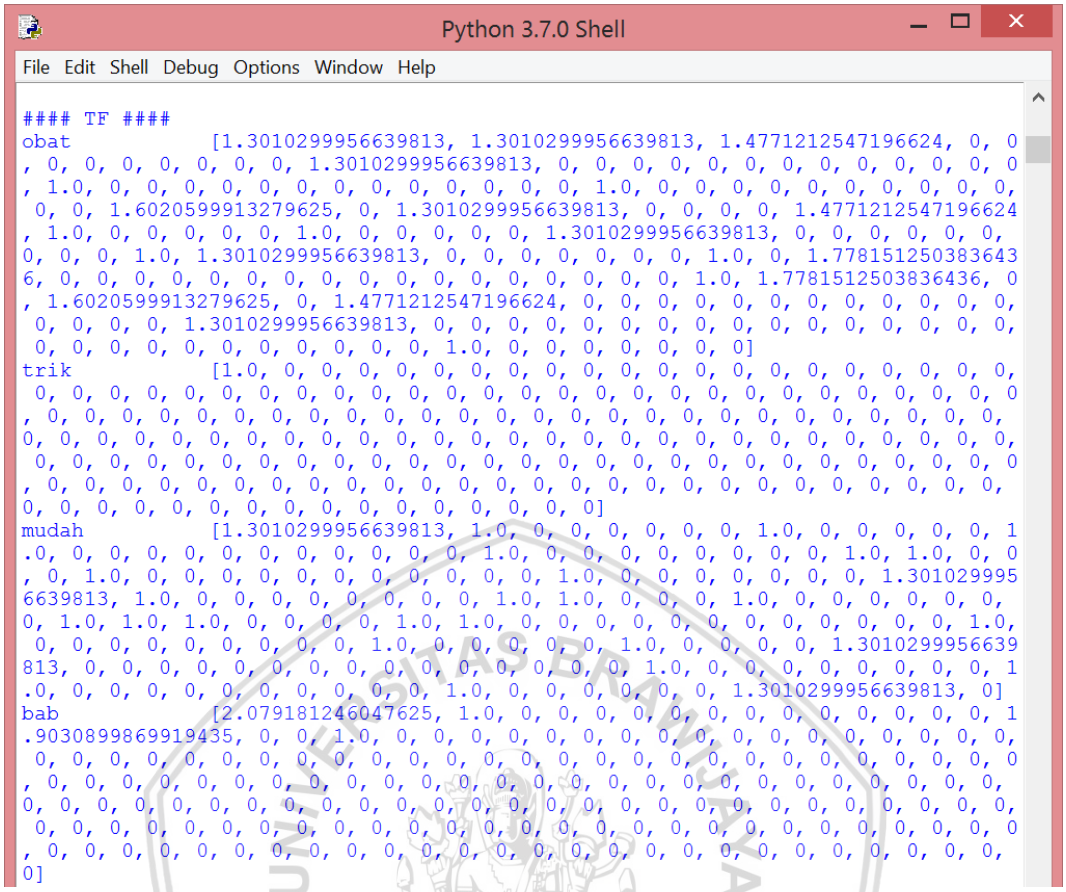
5.3 Implementasi Hasil Pengujian

Implementasi hasil pengujian akan menunjukkan klasifikasi terhadap data uji yang telah dimasukkan sebelumnya. Implementasinya melalui beberapa proses mulai dari *preprocessing*, pembobotan kata, *cosine similarity* data latih hingga klasifikasi menggunakan metode *Modified K-Nearest Neighbor.* Hasil keluaran berupa kalimat yang telah dipecah menjadi kata dan pembuangan kata stopword ditunjukkan pada Gambar 5.1.

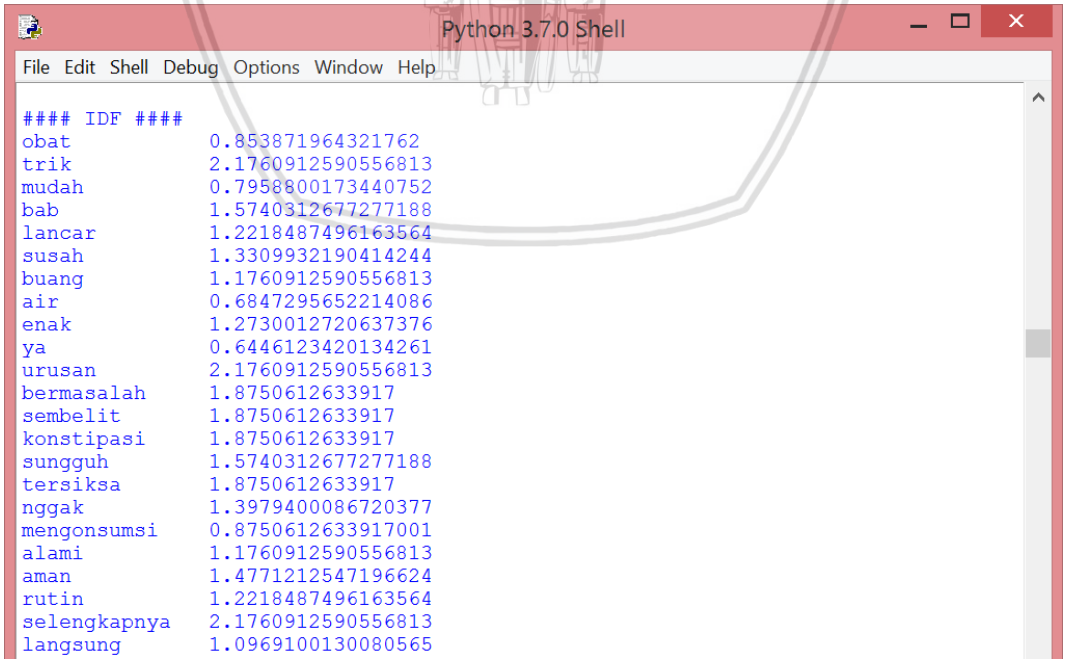


Gambar 5.1 Tampilan Hasil *Preprocessing*

Hasil keluaran berupa TF dari masing-masing kata yang telah dihitung dengan log10 ditunjukkan pada Gambar 5.2 dan IDF dari masing-masing kata ditunjukkan pada Gambar 5.3.

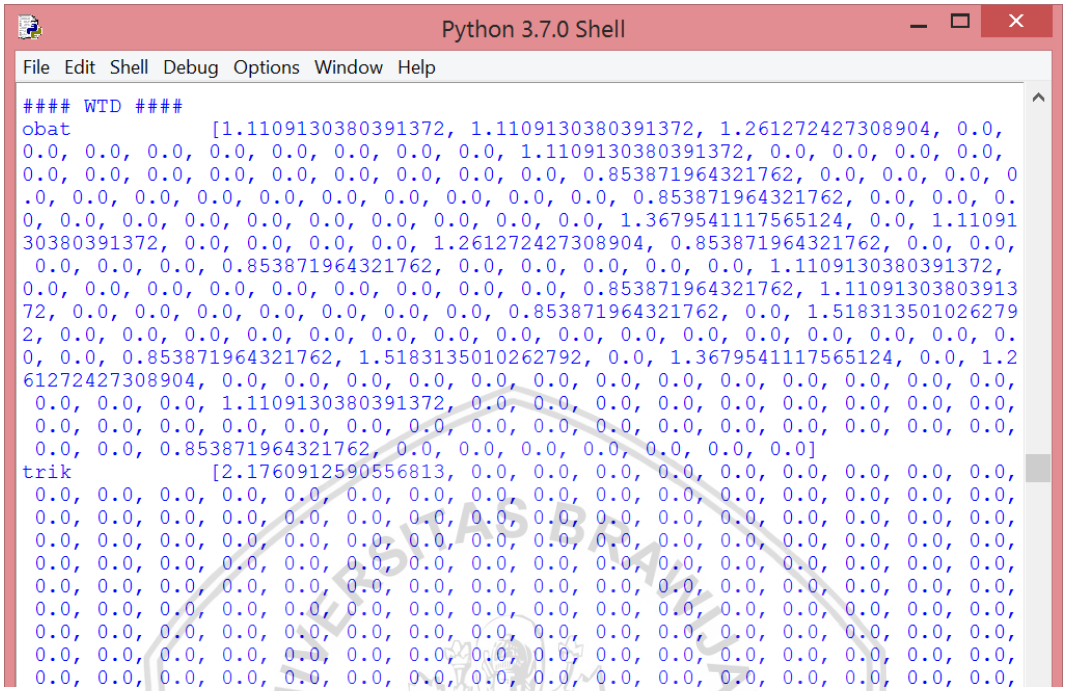


Gambar 5.2 Tampilan Hasil TF

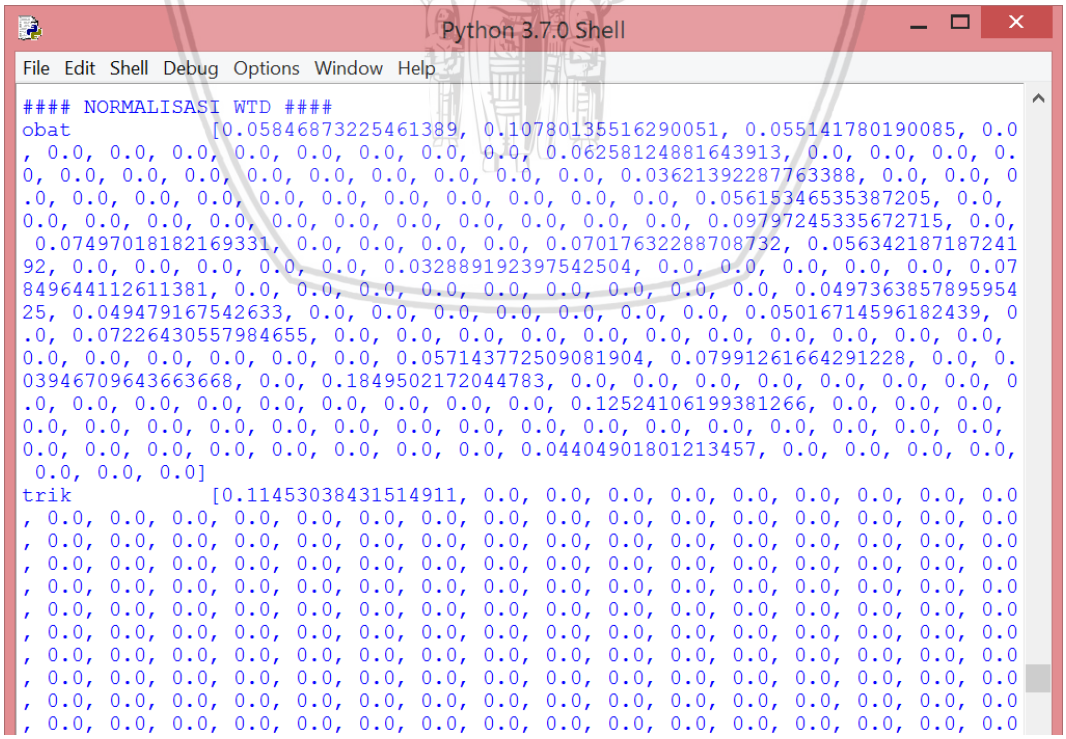


Gambar 5.3 Tampilan Hasil IDF

Hasil keluaran berupa WTD dari masing-masing kata ditunjukkan pada Gambar 5.4 dan normalisasi WTD untuk mempermudah perhitungan *cosine similarity* pada Gambar 5.5.



Gambar 5.4 Tampilan Hasil Perhitungan $W_{t,d}$

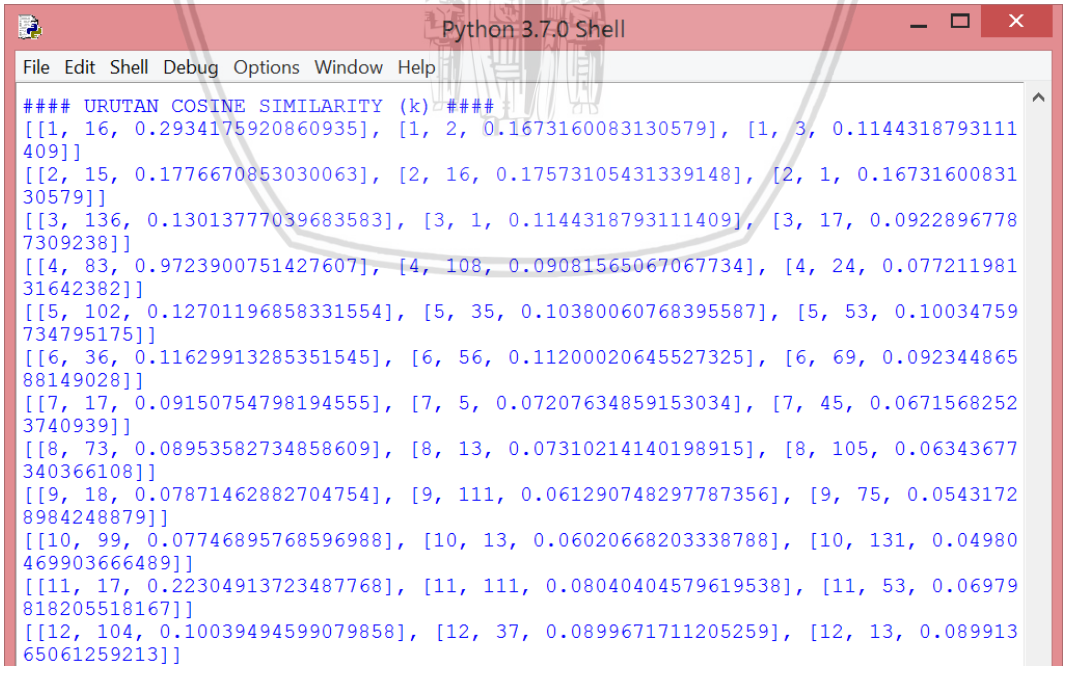


Gambar 5.5 Tampilan Hasil Perhitungan Normalisasi $W_{t,d}$

Hasil keluaran berupa *cosine similarity* antar dokumen data latih ditunjukkan pada Gambar 5.6.

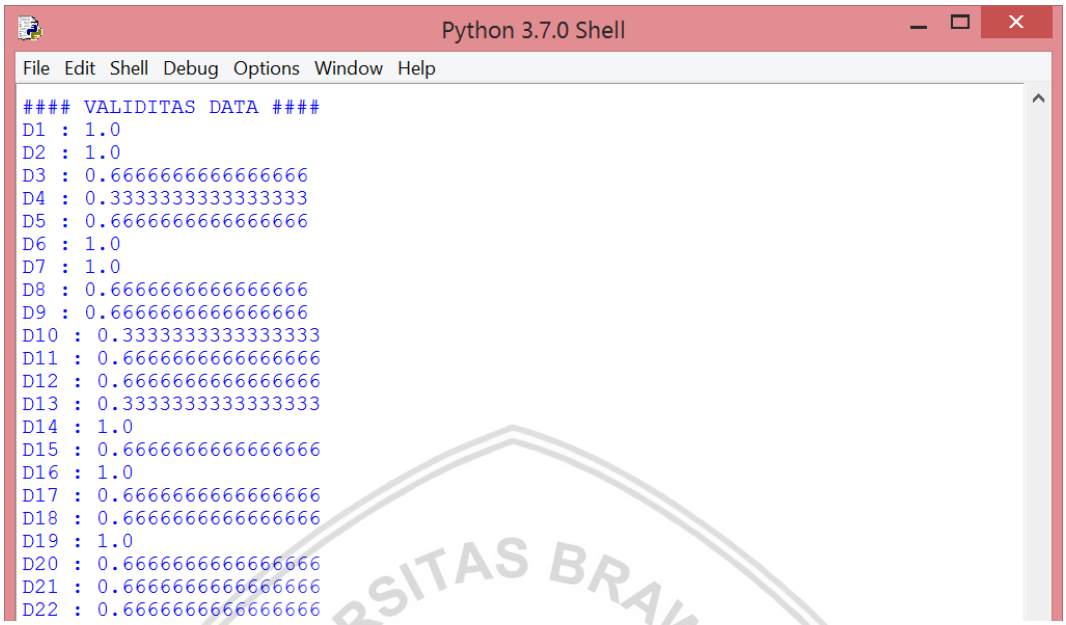
Gambar 5.6 Tampilan Hasil *Cosine Similarity* Data Latih

Hasil keluaran berupa *cosine similarity* antar dokumen data latih ditunjukkan pada Gambar 5.7.



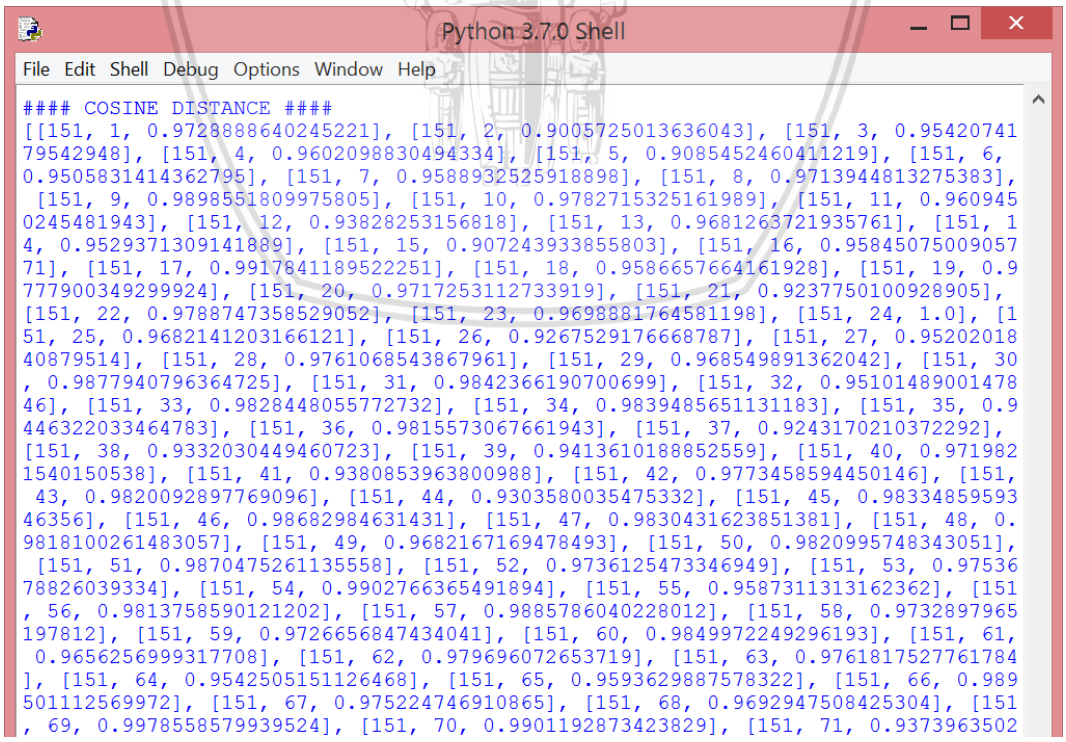
Gambar 5.7 Tampilan Hasil Pengurutan *Cosine Similarity* Data Latih Berdasarkan *K-Values*

Hasil keluaran nilai validitas data antar dokumen data latih ditunjukkan pada Gambar 5.8.



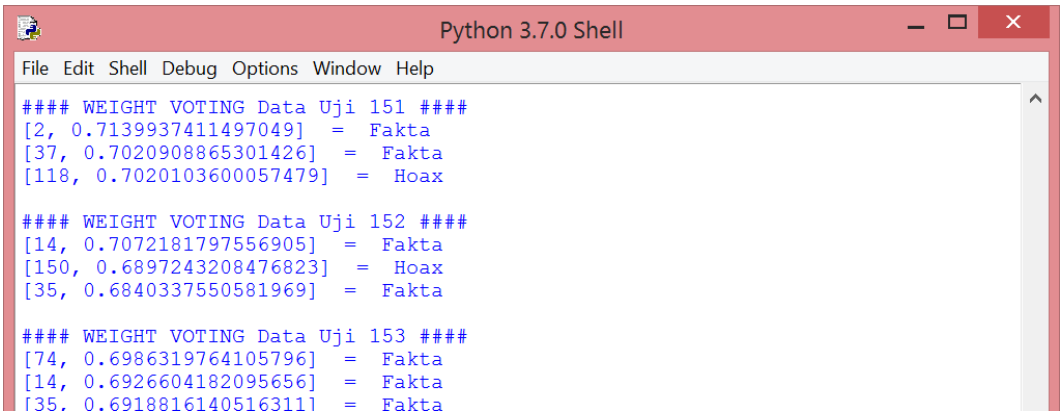
Gambar 5.8 Tampilan Hasil Validitas Data

Hasil keluaran berupa *cosine distance* antar dokumen data latih dengan data uji ditunjukkan pada Gambar 5.9.



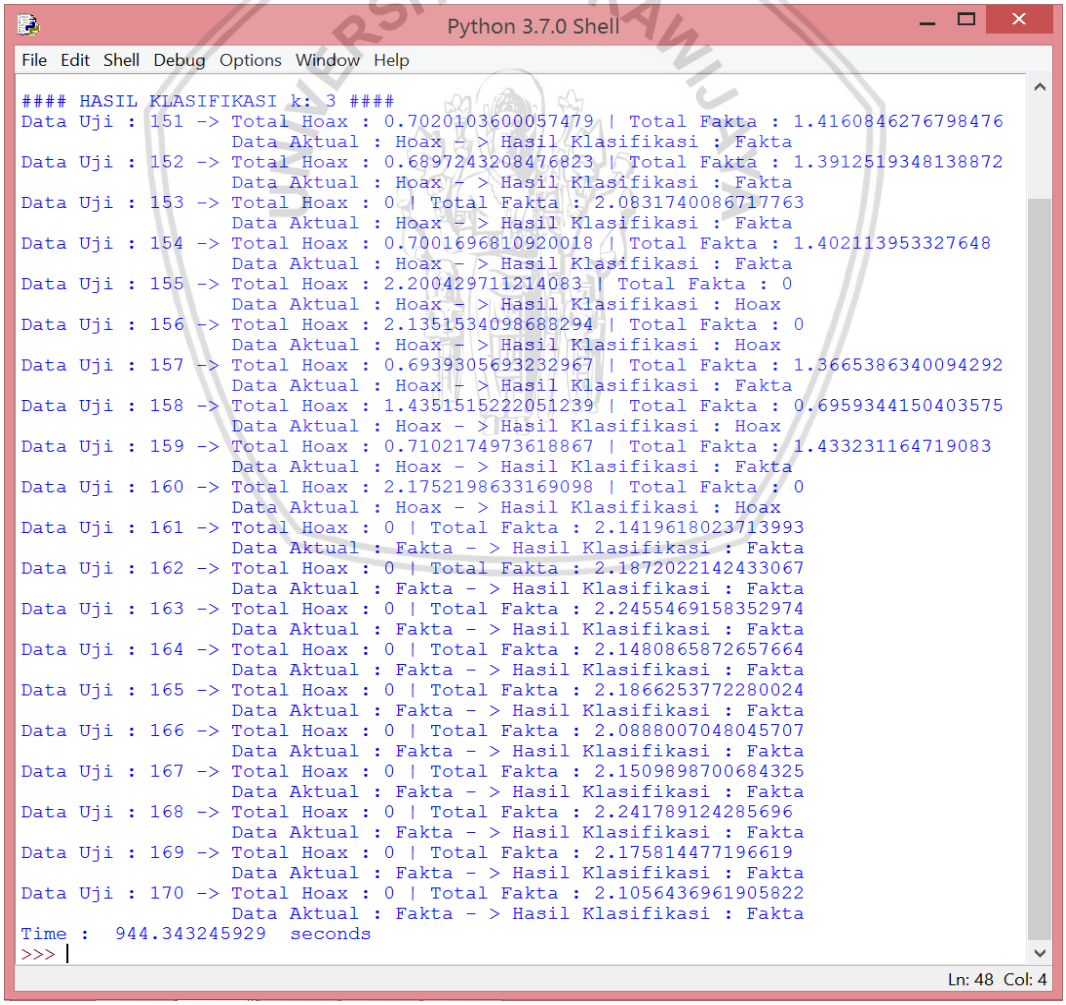
Gambar 5.9 Tampilan Hasil *Cosine Distance*

Hasil keluaran berupa *weight voting* ditunjukkan pada Gambar 5.10.



Gambar 5.10 Tampilan Hasil *Weight Voting*

Hasil keluaran berupa klasifikasi awal dengan klasifikasi oleh sistem ditunjukkan pada Gambar 5.11.



Gambar 5.11 Tampilan Hasil Akhir Klasifikasi Dengan *K-Values* 3

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan membahas mengenai pengujian dari penelitian yang telah dilakukan dan melakukan pembahasan mengenai hasilnya. Masing–masing dari pengujian bertujuan untuk mengetahui tingkat efektivitas klasifikasi. Pengujiannya yakni menggunakan pengujian *k-values* dan pengujian *k-fold cross validation*.

6.1 Pengujian *K-Values*

Untuk mengetahui berapa jumlah tetangga yang menghasilkan akurasi klasifikasi paling tinggi maka dilakukan pengujian menggunakan nilai *k* yang bervariasi yakni 2, 3, 4, 5, 7, 10, 15, 30, 75 dan 100, yang mana data latih berjumlah 150 terdiri dari 75 berita *hoax* dan 75 berita fakta. Untuk data uji menggunakan 10 berita *hoax* dan 10 berita fakta. Hasil pengujian ditunjukkan pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Berdasarkan *K-Values*

K-Values	*Precision*	*Recall*	*F-Measure*	Akurasi
2	0,664835165	0,65	0,657333891	65%
3	0,8125	0,7	0,752066116	70%
4	0,833333333	0,75	0,789473684	75%
5	0,777777778	0,6	0,677419355	60%
7	0,777777778	0,6	0,677419355	60%
10	0,763157895	0,55	0,639278557	55%
15	0,25	0,5	0,333333333	50%
30	0,25	0,5	0,333333333	50%
75	0,25	0,5	0,333333333	50%
100	0,25	0,5	0,333333333	50%

Akurasi tertinggi sebesar 75% terdapat pada pengujian *k-values* yang bernilai 4, dengan nilai *precision* 0,83*,* nilai *recall* 0,75 dan nilai *f-measure* 0,79. Hasil tersebut berdasarkan rata-rata kategori berita *hoax* dan berita fakta. *K-Values* yang paling tinggi tersebut akan dijabarkan hasilnya yakni berupa data aktual serta klasifikasinya pada Tabel 6.2.

Tabel 6.2 Hasil Klasifikasi Data Uji Berdasarkan *K-Values 4*

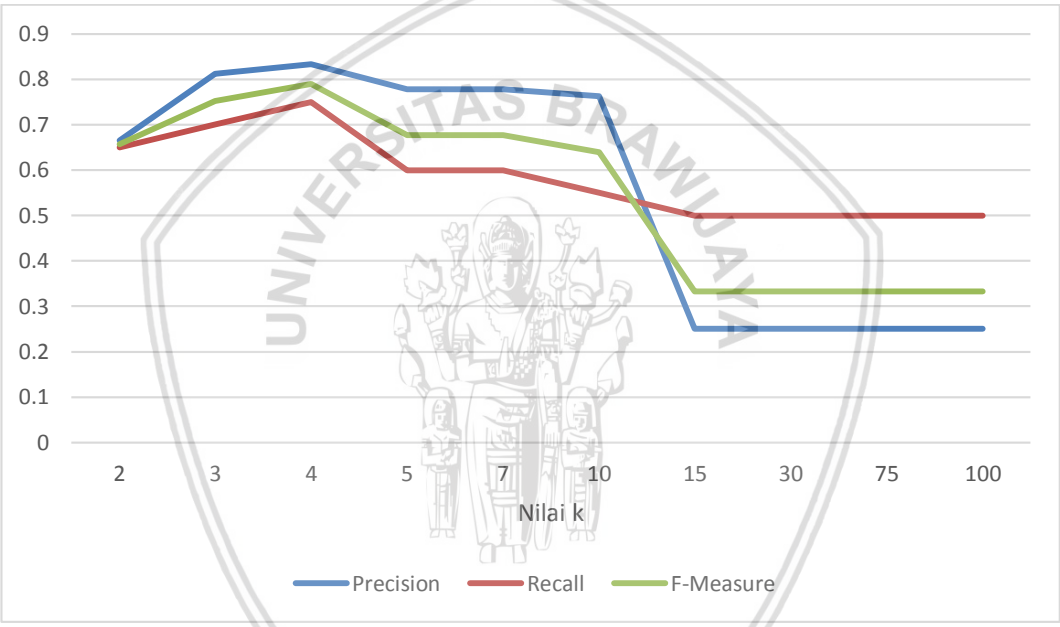
K-Values	Data Uji	Hasil	
		Aktual	**Klasifikasi**
4	D151	*Hoax*	Fakta
	D152	*Hoax*	Fakta
	D153	*Hoax*	Fakta
	D154	*Hoax*	Fakta
	D155	*Hoax*	*Hoax*
	D156	*Hoax*	*Hoax*
	D157	*Hoax*	Fakta
	D158	*Hoax*	*Hoax*
	D159	*Hoax*	*Hoax*
	D160	*Hoax*	*Hoax*
	D161	Fakta	Fakta
	D162	Fakta	Fakta
	D163	Fakta	Fakta
	D164	Fakta	Fakta
	D165	Fakta	Fakta
	D166	Fakta	Fakta
	D167	Fakta	Fakta
	D168	Fakta	Fakta
	D169	Fakta	Fakta
	D170	Fakta	Fakta
Akurasi			75%

Dari 20 data uji yang telah dilakukan terdapat 5 berita *hoax* terklasifikasi benar dan 10 berita fakta terklasifikasi benar. Hasil klasifikasi yang kurang tepat sejumlah 5 berita *hoax* yang mana dimasukkan ke dalam kategori berita fakta. Hasil tersebut dapat dilihat pada tabel *confusion matrix k-values* sebesar 4 yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil *Confusion Matrix* Berdasarkan *K-Values 4*

Hasil Prediksi	Hasil Aktual	
	Hoax	**Fakta**
Hoax	5	0
Fakta	5	10

Tabel selengkapnya untuk hasil klasifikasi data serta *confusion* matrix masing-masing *k-values* dapat dilihat pada Lampiran C. Hasil dari perhitungan evaluasi *precision, recall, f-measure*, klasifikasi data uji dengan menggunakan *confusion matrix* berdasarkan perhitungan seluruh *k-values* ditunjukkan dengan grafik hasil pengujian *k-values* pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian *K-Values*

6.2 Pengujian *K-Fold Cross Validation*

Pengujian ini akan menggunakan 10-*fold* yang mana seluruh data akan dibagi menjadi 10 bagian dan tiap bagiannya berisi 17 data. Untuk kategori fakta pada *fold* 1-5 akan menggunakan data sejumlah 9 dan *hoax* akan menggunakan data sejumlah 8. *Fold* 6-8 akan menggunakan 8 data fakta dan 9 data *hoax* serta seluruh data akan diuji menggunakan *k-kalues* 4. Hasil pengujian *k-fold cross validation* ditunjukkan pada Tabel 6.4.

Tabel 6.4 Hasil 10-*Fold Cross Validation* Berdasarkan *K-Values 4*

K-Values	***Fold***	**Relevan**	**Tidak Relevan**	**Akurasi**
4	*fold* 1	11	6	64,71 %

K-Values	*Fold*	Relevan	Tidak Relevan	Akurasi
4	*fold* 2	9	8	52,94 %
	fold 3	8	9	47,06 %
	fold 4	8	9	47,06 %
	fold 5	12	5	70,59 %
	fold 6	9	8	52,94 %
	fold 7	16	1	94,12 %
	fold 8	12	5	70,59 %
	fold 9	10	7	58,82 %
	fold 10	13	4	76,47 %
Rata-Rata Akurasi				63,53 %

Hasil dari perhitungan 10-*fold cross validation* memperoleh akurasi dengan nilai tertinggi pada *fold* 7 yakni 94,12% yang mana terklasifikasi dokumen yang relevan sebanyak 16 dan yang tidak relevan hanya 1. Untuk rata-rata akurasi mendapatkan hasil 63,53%.

6.3 Perbandingan Dengan Metode *K-Nearest Neighbor*

Perbandingan metode menggunakan *k-values* sama persis yakni dengan nilai 2, 3, 4, 5, 7, 10, 15, 30, 75 dan 100. Jumlah data latih sama sejumlah 150 dan data uji sejumlah 20, masing-masing dengan komposisi data seimbang. Hasilnya ditunjukkan pada tabel 6.5.

Tabel 6.5 Hasil Pengujian Menggunakan Metode *K-Nearest Neighbor*

K-Values	*Precision*	*Recall*	*F-Measure*	Akurasi
2	0.65151515	0.65	0.650756694	65%
3	0.8125	0.8	0.80620155	80%
4	0.8125	0.8	0.80620155	80%
5	0.8125	0.8	0.80620155	80%
7	0.8125	0.8	0.80620155	80%
10	0.85714286	0.8	0.827586207	80%
15	0.33333333	0.75	0.461538462	75%
30	0.35714286	0.8	0.49382716	80%
75	0.35714286	0.8	0.49382716	80%

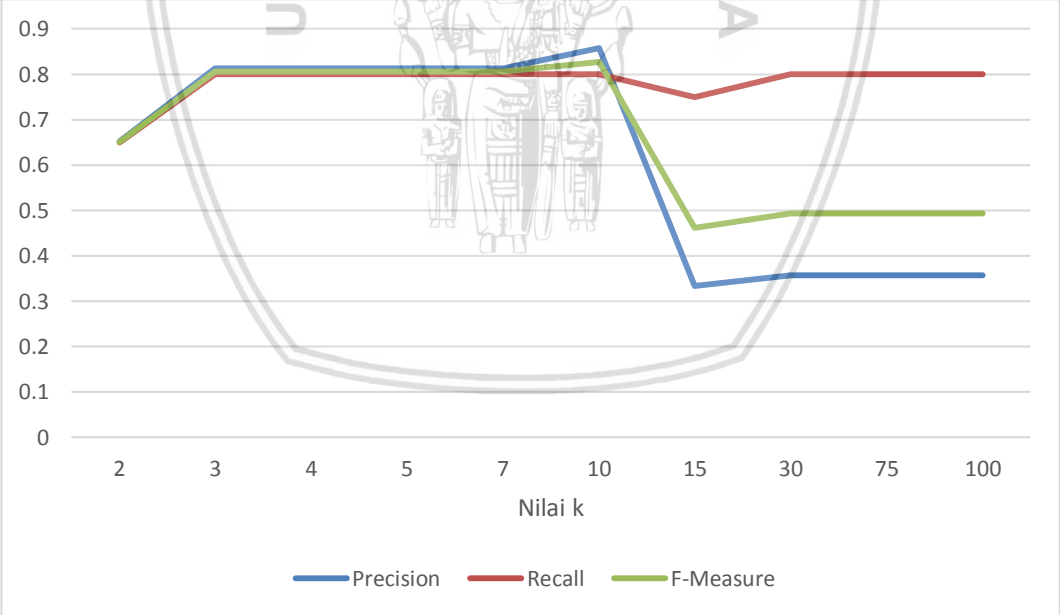
K-Values	*Precision*	*Recall*	*F-Measure*	Akurasi
100	0.35714286	0.8	0.49382716	80%

Akurasi tertinggi sebesar 80% terdapat pada pengujian *k-values* dengan menggunakan metode *K-Nearest Neighbor* yang bernilai 10, dengan nilai *precision* 0,86 nilai *recall* 0,8 dan nilai *f-measure* 0,83. Hasil tersebut berdasarkan rata-rata kategori berita *hoax* dan berita fakta. Tabel *confusion matrix* ditunjukkan pada Tabel 6.6.

Tabel 6.6 *Confusion Matrix* Pada Metode *K-Nearest Neighbor*

Hasil Prediksi	Hasil Aktual	
	Hoax	**Fakta**
Hoax	5	0
Fakta	5	10

Hasil dari perhitungan evaluasi *precision, recall, f-measure*, klasifikasi data uji dengan menggunakan *confusion matrix* berdasarkan perhitungan seluruh *k-values* ditunjukkan dengan grafik hasilnya pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Pengujian Menggunakan Metode *K-Nearest Neighbor*

Sistem yang menggunakan metode *K-Nearest Neighbor* dengan *k-values* bernilai 10 dapat menghasilkan hasil klasifikasi lebih baik yakni dengan 6 berita *hoax* dapat terklasifikasi dengan benar sementara untuk berita fakta berhasil terklasifikasi dengan benar sejumlah 10 berita. Akurasi yang didapatkan sebesar 80%. Dapat disimpulkan bahwa klasifikasi berita kesehatan berbahasa Indonesia

dengan menggunakan metode *K-Nearest Neighbor* lebih baik daripada menggunakan metode *Modified K-Nearest Neighbor.*

6.4 Analisis

Pengujian *k-values* bernilai 4 menghasilkan akurasi tertinggi, sistem dapat menunjukkan hasil akurasi terbaik karena sistem ini menghitung berdasarkan frekuensi kata, jadi hasilnya akan sangat bergantung dari banyaknya kata kunci yang sama dari masing-masing kategori berita. Dengan hanya beberapa data latih saja hal tersebut sudah mewakili karakteristik dari data uji dikarenakan kemiripan antar kata. Akurasi hanya sebesar 75% juga diakibatkan banyaknya berita *hoax* yang menggunakan kata-kata yang mirip berita fakta, hal tersebut bertujuan untuk meyakinkan pembaca dalam artian lain kata yang dimasukkan ke dalam konten berita *hoax* disusun sedemikian rupa. Banyaknya kata singkatan dan kata tidak baku juga berpengaruh yang mana bisa dianggap berbeda makna.

Pengujian *k-fold* mendapat akurasi tertinggi pada *fold* 7 yang mana menunjukkan akurasi paling tinggi dikarenakan yang pertama terdapat kata-kata dalam berita hoax tersebut hanya terdapat pada data tersebut sendiri. Kedua dikarenakan banyak dari kata yang terkandung dalam data uji didominasi dari kata yang mempunyai kategori serupa. Contohnya berita kategori *hoax* dengan kata kunci "anthrax" hanya terdapat pada 1 berita *hoax* saja. Konten berita *hoax* yang terlalu sedikit juga dapat mengakibatkan salah dalam mengklasifikasikan berita tersebut. Sedangkan berita fakta rata-rata menggunakan kata-kata yang baku dan variasi kata serta frekuensi katanya pun banyak.

Perbandingan sistem yang menggunakan *Modified K-Nearest Neighbor* dengan metode *K-Nearest Neighbor* mendapatkan hasil akurasi yang lebih tinggi jika menggunakan *K-Nearest Neighbor* saja dikarenakan hanya menggunakan perhitungan *cosine similarity* antara data latih dengan data uji, lalu hasilnya diurutkan mana yang mempunyai nilai kemiripan tertinggi. Jumlah kata yang mempunyai frekuensi terbanyak akan sangat berpengaruh karena kata yang dibandingkan antar dokumen adalah seluruh kata tanpa pengkhususan kata.

Dari perbandingan metode menggunakan pengujian *k-values* yakni metode *Modified K-Nearest Neighbor* dan *K-Nearest Neighbor* dapat disimpulkan jika *k-values* dalam rentang angka 1 sampai 10 akan menghasilkan hasil klasifikasi yang baik, sementara untuk rentang angka 10 keatas hasil klasifikasi sudah tidak baik lagi. Berita fakta lebih sering mendapatkan hasil klasifikasi yang mendekati sempurna dikarenakan kata yang terkandung di dalam berita tersebut lebih banyak porsinya, sementara pada berita *hoax* ada beberapa yang memang mempuyai konten dengan jumlah kata sedikit.

BAB 7 PENUTUP

Bab ini akan membahas mengenai kesimpulan dan saran dari hasil penelitian Klasifikasi *Hoax* Pada Berita Berbahasa Indonesia Dengan Menggunakan Metode *Modified K-Nearest Neighbor*:

7.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian ini adalah sebagai berikut:

1. Metode *Modified K-Nearest Neighbor* dapat digunakan pada sistem klasifikasi berita dengan input yang berupa teks lalu diproses menjadi beberapa tahapan mulai dari *preprocessing*, pembobotan kata, *cosine similarity* antar data latih, validitas data antar data latih untuk mengetahui tingkat kemiripan kategori berita, *cosine distance* dan *weight voting* untuk mengurutkan dan menentukan klasifikasi yang terbesar berdasarkan *k-values.*
2. Hasil sistem yang menggunakan metode *K-Nearest Neighbor* mendapatkan akurasi lebih tinggi yakni sebesar 80%.
3. Hasil dari pengujiannya yakni akurasi tertinggi sebesar 75% terdapat pada pengujian *k-values* yang bernilai 4, dengan nilai *precision* 0,83, nilai *recall* 0,75 dan nilai *f-measure* 0,79. Hasil klasifikasi mempunyai akurasi tidak terlalu tinggi dikarenakan topik dari teks berita kesehatan masih terlalu umum serta terdapat banyak teks yang kurang baku atau berupa kata singkatan sehingga sistem sulit untuk mengklasifikasikan dengan tepat.
4. Pengujian *k-fold* dengan *fold* 7 menunjukkan akurasi paling tinggi yakni sebesar 94,12%. Hal tersebut dikarenakan ada 1 berita memiliki konten "anthrax" yang hanya terdapat pada data itu.

7.2 Saran

Membutuhkan data yang lebih banyak lagi dari klasifikasi yang dilakukan oleh pakar. Kata-kata yang terkandung dalam konten berita dari kategori *hoax* banyak yang kurang baku serta banyak kata yang bermakna berbeda namun dengan kata yang sama dihitung dalam frekuensi kata yang sama itu sendiri, hal tersebut menyebabkan sistem salah dalam mengklasifikasikannya dan sebaiknya data dinormalisasi dahulu agar memperoleh hasil yang optimal.

Sistem yang telah dikembangkan menggunakan metode *Modified K-Nearest Neighbor* ini kurang efisien dengan jumlah data latih yang memiliki konten terlalu umum oleh sebab itu dapat dilakukan pengembangan kedepannya dengan penambahan data latih yang mencakup seluruh topik kesehatan atau bisa juga dengan mengkolaborasikan metode *Modified K-Nearest Neighbor* dengan metode lainnya yang diharapkan dapat memberikan klasifikasi lebih baik lagi.

DAFTAR PUSTAKA

Adikara, P. P., Perdana R. S. dan Indriati, 2017. *Model Vector Space*. Malang: Fakultas Ilmu Komputer Universitas Brawijaya.

Abrar, A. N., 2005. *Penulisan berita*. Yogyakarta: Universitas Atma Jaya.

Amelia, M., 2016. *Selama 2016, 300 Akun Medsos Penyebar Hoax Diblokir Polisi*. [online] Tersedia di: <http://news.detik.com/berita/d-3384819/selama-2016-300-akun-medsos-penyebar-hoax-diblokir-polisi> [Diakses 14 Februari 2018]

Arnaz, F., 2016. *Hoax di Medsos, Polri: Pelakunya Sekarang Hit and Run*. [online] Tersedia di: <http://www.beritasatu.com/hukum/406905-hoax-di-medsos-polri-pelakunya-sekarang-hit-and-run.html> [Diakses 17 Februari 2018]

Anggono, R., Arie A. S., dan Angelina P. K., 2009. *Analisis perbandingan metode K-Nearest Neighbor dan Naïve Bayes classifier dalam klasifikasi teks*. S1. Universitas Telkom. Tersedia di: <http://repository.telkomuniversity.ac.id/pustaka/94560/analisis-perbandingan-metode-k-nearest-neighbor-dan-naive-bayes-classifier-dalam-klasifikasi-teks.html> [Diakses 13 Maret 2018]

Bisono, E. F., 2010. Penentuan status tahapan keluarga sejahtera dengan menerapkan metode Modified K-Nearest Neighbor. S1. Universitas Brawijaya.

Cahya, I. S., 2012. *Menulis berita di media massa*. Yogyakarta: Citra Aji Pratama.

Chaer, A., 2013. *Pengantar semantik bahasa Indonesia*. Jakarta: Reneka Cipta.

Chen, Y. Y., Suet-Peng Y., dan Adzlan I., 2014. Email hoax detection system using levenshtein distance method. *Journal of computers*, (2)9, pp.441-446.

Dahlan, M. A., 2017. *Ahli: 'Hoax' Merupakan Kabar Yang Direncanakan*. [online] Tersedia di: <https://www.antaranews.com/berita/606085/ahli-hoax-merupakan-kabar-yang-direncanakan> [Diakses 4 April 2018]

Fauzi, A. M., 2017. *Text Mining 2017/2018*. [online] Tersedia di: < http://malifauzi.lecture.ub.ac.id/2017/09/text-mining-20172018/> [Diakses 4 April 2018]

Hardiyanti, S., 2014. Implementasi Metode Modified K-Nearest Neighbor (MKNN) Pada Penentuan Keminatan Sekolah Menengah Atas (SMA) (Studi Kasus: SMA Negeri 1 Seririt). S1. Universitas Brawijaya.

Herwijayanti, B., Ratnawati D. E. dan Muflikhah L., 2018. Klasifikasi berita online dengan menggunakan pembobotan tf-idf dan cosine similarity. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, (1)2, p.307.

Laoly, Y. H., 2016. *Serbuan 10 Juta TKA Asal Cina, Menkumham: Itu Hoax!*. [online] Tersedia di: <http://www.republika.co.id/berita/nasional/umum/

16/12/29/oixl41361-serbuan-10-juta-tka-asal-cina-menkumham-itu-hoax> [Diakses 4 April 2018]

Nathania, D. Z., 2017. Klasifikasi spam pada Twitter menggunakan Improved K-Nearest Neighbor. S1. Universitas Brawijaya.

Palinoan, V. W. dan Wijono S. H., 2014. *Sistem klasifikasi dokumen bahasa jawa dengan metode K-Nearest Neighbor (K-NN)*. S1. Universitas Sanata Dharma. Tersedia di: <https://repository.usd.ac.id/4346/> [Diakses 21 Maret 2018]

Parvin, H., Alizadeh, H., dan Bidgoli, B., 2008. MKNN: Modified K-Nearest Neighbor. *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco: USA.

Parvin, H., Alizadeh, H., dan Minati, B., 2010. A Modification on K-Nearest Neighbor Classifier. *Global Journal of Computer Science and Technology*, (1)10, pp.38.

Prasetyo, Y. A., 2017. *Ini Hoax Versi Dewan Pers*. [online] Tersedia di: <https://kriminologi.id/hard-news/cyber-crime/ini-hoax-versi-dewan-pers> [Diakses 4 April 2018]

Rasywir, E. dan Ayu P., 2015. Eksperimen pada sistem klasifikasi berita hoax berbahasa indonesia berbasis pembelajaran mesin. *Jurnal Cybermatika*, (2)3, pp.1-8.

Setya, A., 2016. *Hoax di Medsos, Polri: Pelakunya Sekarang Hit and Run*. [online] Tersedia di: < http://www.beritasatu.com/hukum/406905-hoax-di-medsos-polri-pelakunya-sekarang-hit-and-run.html> [Diakses 4 April 2018]

Sumadiria, A. H., 2005. *Jurnalistik Indonesia: menulis berita dan feature: panduan praktis jurnalis professional*. Bandung: PT. Remaja Rosdakarya.